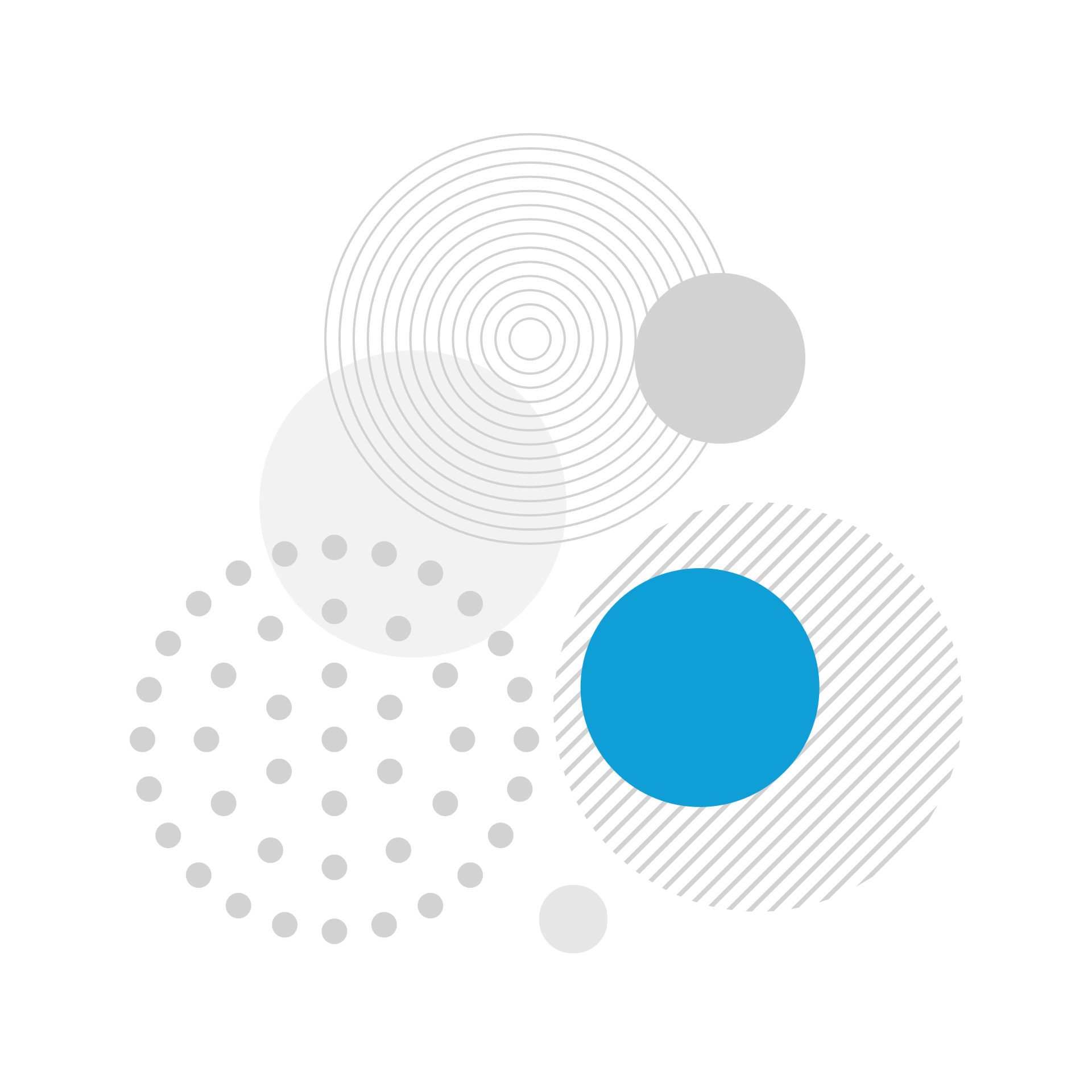
# **TRƯỜNG ĐẠI HỌC FPT KHOA KĨ THUẬT PHẦN MỀM**

A black background with colorful text

Description automatically generated



Professor: Ms. Do Thi Loan

Class: SE1849

Student: Le Trong Linh

Student ID: HE182116

Hanoi, 8/2024

Unimart E-Commerce

Projectt

**TABLE OF CONTENTS**

[**TRƯỜNG ĐẠI HỌC FPT KHOA KĨ THUẬT PHẦN MỀM** 1](#_Toc160800678)

[**1. TECHNOLOGIES USED** 5](#_Toc160800679)

[**1.1 Frontend** 5](#_Toc160800680)

[***1.1.1 HTML (HyperText Markup Language)*** 5](#_Toc160800681)

[***1.1.2 CSS (Cascading Style Sheets)*** 7](#_Toc160800682)

[***1.1.3 JavaScript*** 11](#_Toc160800683)

[**1.2 Backend** 13](#_Toc160800684)

[***1.2.1 Java with Apache Tomcat*** 13](#_Toc160800685)

[**1.3 Database** 16](#_Toc160800686)

[***1.3.1 SQL Server*** 16](#_Toc160800687)

[***1.3.2 MongoDB*** 19](#_Toc160800688)

[**2. SURVEY, ANALYSIS AND SYSTEM DESIGN** 21](#_Toc160800689)

[**2.1 Survey** 21](#_Toc160800690)

[***2.1.1 Introduction to the Unimart Web Application*** 21](#_Toc160800691)

[***2.1.2. Current Assessment*** 21](#_Toc160800692)

[***2.1.3. Purpose of website*** 22](#_Toc160800693)

[***2.1.4. Requirement*** 22](#_Toc160800694)

[**2.2 Analysis and System Design** 23](#_Toc160800695)

[***2.2.1. Detailed Requirements for the Website*** 23](#_Toc160800696)

[***2.2.2. Diagram*** 24](#_Toc160800697)

[***2.2.3. Database Design*** 30](#_Toc160800698)

[**3. WEBSITE INTERFACE** 34](#_Toc160800699)

[***3.1 Home Page*** 34](#_Toc160800700)

[***3.2 Login Page*** 35](#_Toc160800701)

[***3.3 Sign Up Page*** 35](#_Toc160800702)

[***3.4 Collections Page*** 36](#_Toc160800703)

[***3.5 Product Detail Page*** 37](#_Toc160800704)

[***3.6 Cart Page*** 38](#_Toc160800705)

[***3.7 Order Page*** 39](#_Toc160800706)

[***3.8 Profile Page*** 40](#_Toc160800707)

[***3.9 Add Product Page***  41](#_Toc160800708)

[***3.10 Order History Page*** 42](#_Toc160800709)

[**Conclusion:** 42](#_Toc160800710)

**PREFACE**

In the wake of the digital era, e-commerce has emerged as a transformative force across the globe, reshaping economic landscapes and consumer behaviors. Amid this digital revolution, Vietnam stands at the forefront, embracing the shift towards online marketplaces that promise convenience, diversity, and quality.

Recognizing the unique needs and challenges faced by university students, UniMart has been developed as a specialized e-commerce platform. Its purpose is clear: to facilitate the exchange of goods within the vibrant ecosystem of a university campus. This innovative platform is not merely a marketplace but a community-centric solution designed to make campus life more convenient and sustainable.

UniMart distinguishes itself by focusing exclusively on the needs of the student population. It offers a dedicated space for students to buy, sell, or exchange items, ranging from textbooks to electronics, within the safety and comfort of their campus. This approach not only simplifies transactions but also fosters a culture of sustainability and responsible consumption among the youth.

The creation of UniMart reflects a broader vision: to leverage digital tools for solving real-world problems. By focusing on the principles of a circular economy, the platform aims to minimize waste and encourage the reuse of goods, contributing to a more sustainable and environmentally conscious campus environment.

As UniMart launches into Vietnam's dynamic e-commerce landscape, it aspires to be more than just a transactional space. It seeks to embody the values and lifestyles of modern university students, offering innovative solutions that are practical and aligned with the ideals of sustainability and community building. Through this initiative, UniMart is set to redefine the campus experience, making every exchange an opportunity for positive impact and connection.

# **1. TECHNOLOGIES USED**

## **1.1 Frontend**

### ***1.1.1 HTML (HyperText Markup Language)***

HTML (HyperText Markup Language) is a widely used markup language for constructing and structuring web pages. This language allows programmers to define the structure and display content on web browsers clearly. HTML utilizes syntax based on tags, where each tag marks a specific part of the page, and tag pairs enclose the content to be applied.

Each HTML page typically begins with a doctype declaration, followed by a basic structure with main tags such as <html>, <head>, and <body>. The <head> section contains meta information about the page, title, and links to CSS or JavaScript files. The <body> section contains the content displayed on the browser.

HTML tags are diverse and rich, allowing the insertion of text, images, links, and other components into web pages. Each tag can be expanded by adding attributes to customize or provide additional information. For example, the <a> tag is used to create links and has an “href” attribute to specify the path.

HTML continues to evolve, with new versions like HTML5 bringing many new features, multimedia support, and improved interactivity. This language plays a crucial role in web development, shaping how information is transmitted and displayed on the internet.

#### **1.1.1.1 Definition of Basic HTML Tags**

Tags are keywords placed inside angle brackets, each tag serving a distinct function:

* The <p> tag is used to create a paragraph of text.
* The <a> tag is used to create a hyperlink.
* The <img> tag is used to insert an image into a web page…
* Tags are typically divided into two main types: paired tags & singleton tags
* Paired tags are tags that when we want to use them to create content for the web page, we need to use both opening and closing tags (the closing tag is written similarly to the opening tag, except that before the tag name, there is an additional slash - for example: <tagname>displayed content</tagname>.
* Singleton tags are tags that when we want to use them to create content for the web page, we only need to use the opening tag, or in other words, there is no closing tag. For example, the <img> tag belongs to the singleton tag category, used to insert an image into the web page.
* Các thẻ tags cơ bản trong HTML:
* <html>: This tag marks the beginning of a web page and wraps around the entire content.
* <head>: Contains meta information about the page, title (<title>), links to CSS, JavaScript files, and other infomation.
* <body>: Contains meta information about the page, title (<title>), links to CSS, JavaScript files, and other information.
* <h1>, <h2>, ..., <h6>: Used to create headings with decreasing size from large to small.
* <p>: Marks a paragraph of text.
* <a>: Creates a link to another web page. It has an href attribute to specify the path.
* <img>: Inserts an image into the page. It has an src attribute to specify the image path.
* <ul>, <ol>, <li>: <ul> creates an unordered list, <ol> creates an ordered list, and <li> represents each item in the list.
* <div>: Often used to group and organize content within a block.
* <span>: Similar to <div>, but often used to apply styles or change formatting for a small portion of text.
* <form>:  Creates a form for users to input information. Various child tags like <input>, <textarea>, and <button> are used within <form>.
* <input>: Used in forms to create input fields, radio buttons, checkboxes, and various other types of inputs.
* <table>: Creates a table for organizing data. Tags like <tr> (rows), <td> (data cells), <th> (header cells) are commonly used within <table>.
* <br>: Creates a new line in text.
* <hr>: Creates a horizontal line to divide content.
* <textarea>: The Textarea tag allows users to input data with multiple lines. With this tag, you cannot limit the maximum length on the Web page.
* <select>: The Select tag allows users to select elements from a predefined set. If the Select tag allows users to select an element from the list of elements, it will function like a combobox. If the Select tag allows users to select multiple elements at once from the list of elements, that Select tag is a listbox type.
* <form>: When you want to submit data entered by users from the Client side of the Web to the Server side, you have two ways to do it corresponding to two methods POST and GET in the form tag. Each form tag will declare an action pointing to a different page.

### ***1.1.2 CSS (Cascading Style Sheets)***

**CSS (Cascading Style Sheets) is a language used to locate and format elements generated by markup languages (HTML), or in other words, to create styles for web pages. While HTML serves to format elements on the website such as creating paragraphs, headers, tables, etc., CSS adds style to these HTML elements such as changing layouts, page colors, text colors, fonts, and structural alterations.**

**CSS was developed by the W3C (World Wide Web Consortium) in 1996 because HTML was not designed to attach tags to help format web pages.**

**The working method of CSS is that it will search based on selectors, selectors can be the name of an HTML tag, the name of an ID, a class, or many other types. Then it will apply the necessary property changes to that selector.**

**CSS can be declared in many different ways. CSS snippets can be placed within the <style> tag, written into a separate file with the extension ".css", or placed within individual HTML tags. However, depending on the placement method, their priority levels also differ. Below is a basic priority level system:**

**1. !important: The highest priority level is given to rules with the !important property. However, using !important is not recommended as it can make CSS code difficult to understand and maintain.**

**2. Inline Styles: If a style is applied directly to an element through the style attribute, it will have a higher priority level than other CSS rules.**

**3. ID Selectors: ID selectors (#example) have a higher priority level than class or element selectors. However, it's advisable to avoid using too many IDs in CSS to avoid complexity and maintenance issues.**

**4. Class Selectors, Attributes, and Pseudo-classes: Class selectors (.example), attribute selectors ([type="text"]), and pseudo-classes (:hover, :first-child, etc.) have the same priority level and lower priority than IDs.**

**5. Type Selectors: Element type selectors (div, p, h1, etc.) have the lowest priority level. If there are no selectors with higher priority, the rule applies to all elements of that type. *1.1.2.1 Definition of Basic about CSS***

***Definitions of some font-related properties***

|  |  |  |
| --- | --- | --- |
| No | Attribute | Description |
| 1 | font-size: 12px  font-size: 2em | Sets the font size using either pixels or em units. |
| 2 | font-family: Arial, Helvetica, sans-serif | Specifies the font family for text. |
| 3 | font-weight: bold | Makes the text bold. |
| 4 | font-style: italic | Makes the text italicized. |
| 5 | text-decorator: underline | Adds an underline to the text. |
| 6 | text-transform: uppercase | Converts the text to uppercase. |
| 7 | color: red | Sets the color of the text to red. |

***Definitions of some object-related properties***

|  |  |  |
| --- | --- | --- |
| No | Attribute | Description |
| 1 | width: 100px | Sets the width of the object to 100 pixels. |
| 2 | height: 200px | Sets the height of the object to 200 pixels. |
| 3 | background: url(images/bg.png) | Sets the background image of the object to ‘bg.png’ located in the ‘images’ folder. |
| 4 | background: #f0f0f0  background: red | Sets the background color of the object to either a hexadecimal color (‘#f0f0f0’) or a color name (‘red’). |
| 5 | border: 1px dashed black | Sets a border around the object with a width of 1 pixel, a dashed style, and a color of black. |
| 6 | margin: 10px 5px 8px 7px | Sets the margin space around the object. The values represent the top, right, bottom, and left margins respectively. |
| 7 | padding: 10px 5px 8px 7px | Sets the padding space inside the object. The values represent the top, right, bottom, and left padding respectively. |

***Other properties***

|  |  |  |
| --- | --- | --- |
| No | Attribute | Description |
| 1 | display: flex | Turns the object into a flexbox, allowing flexible arrangement, alignment, etc., of child elements. |
| 2 | transform: scale(1.3) | Scales the size of the object. The value 1.3 scales it to 130% of its original size. |
| 3 | transition: transform 0.3s ease | Creates a smooth transition effect for the element when the ‘transform’ attribute changes. The transition lasts for 0.3 seconds and has an ease timing function. |
| 4 | box-sizing: border-box | Treats padding and border as part of the object's height and width. This ensures that the declared width and height include padding and border, rather than being added to them. |

### ***1.1.3 JavaScript***

JavaScript is a widely used and important programming language in the field of web development. Initially created to provide dynamic interactivity for web pages, JavaScript has now become a versatile, powerful, and multitasking language used not only in web browsers but also across various platforms.

JavaScript allows web developers to create dynamic, engaging, and interactive applications without the need to reload pages. It supports multiple programming paradigms, including functional programming, object-oriented programming, and control structure programming. This flexibility enables JavaScript to evolve and adapt to various types of projects and different requirements.

This language is also widely used in both front-end and back-end development of web applications through various frameworks. This creates an integrated web development ecosystem, helping developers build efficient and robust web applications.

#### **1.1.3.1 Definition of Basic about JavaScript**

***Variables:*** Variables are used to store and reference data within a program.

* To declare variables, you can use the keywords var, let, or const:
* var age = 25;
* let username = "John";
* const PI = 3.14;

***Functions:*** Functions are blocks of code that perform a specific task.

- To declare a function, you use the function keyword:

function greet(name) {

console.log("Hello, " + name + "!");

}

***Events:*** JavaScript allows handling events when users interact with a web page.

- Events may include click, hover, submit, and many other events.

- Example:

document.getElementById("myButton").addEventListener("click", function() {

alert("Button clicked!");

});

***Objects:*** JavaScript is an object-oriented language, and objects are a way to organize code into groups of attributes and related methods.

- Example:

var car = {

brand: "Toyota",

model: "Camry",

start: function() {

console.log("Engine started!");

}

};

**Arrays:** Arrays are a data structure to store multiple values in a single variable.

- Example:

var colors = ["red", "green", "blue"];

***DOM (Document Object Model):*** DOM is a representation of an HTML, XML, or XHTML document as a tree structure. JavaScript uses DOM to interact with and modify the structure and content of a web page.

- Example:

document.getElementById("myElement").innerHTML = "New content";

## **1.2 Backend**

### ***1.2.1 Java with Apache Tomcat***

Java is a versatile, powerful, and widely used programming language developed by Sun Microsystems (now part of Oracle Corporation). First introduced in 1995, Java has become one of the most popular programming languages globally, widely used in various fields.

Java source code can run on any platform that supports the Java Virtual Machine (JVM) without needing to recompile the source code for each specific operating system. This makes Java platform-independent and easily transferable. Additionally, Java has a secure and safe runtime environment. Some security features of Java include automatic memory management, object access control, and restrictions on pointer usage.

Java is designed based on the object-oriented model, with full support for concepts such as class, object, inheritance, and polymorphism. Java also provides a rich standard library (Java Standard Edition - JSE) to reduce development time and increase application flexibility. This library includes many packages to support tasks such as file I/O, networking, string handling, and much more.

Java is widely used in web application development through frameworks like Spring and JavaServer Faces (JSF). Moreover, Java is primarily used in mobile application development through the Android SDK. Therefore, the Java programming community is a strong and diverse community. This community not only provides support but also contributes to the development of powerful libraries and frameworks such as Spring, Hibernate, and Apache Struts.

Java plays a crucial role in the software industry, from desktop application development to mobile programming and web development. This demonstrates the strength and flexibility of Java in meeting the diverse needs of the programming community.

#### **1.2.1.1 Giới thiệu về thư viện JavaServer Pages Standard Tag Library (JSTL)**

JSTL (JavaServer Pages Standard Tag Library) is a standard tag library for the JavaServer Pages (JSP) language. It provides a set of tags to perform common tasks in web programming, helping to reduce source code complexity and increase code reuse in Java applications.

***Features of the JSTL Library***

|  |  |  |
| --- | --- | --- |
| STT | Tag and tag group | Description |
| 1 | - <c:if test=“”>  - <c:choose test =“”>  <c:when test =“”>  <c:otherwise test =“”> | Checks the condition in the "test" attribute to execute the function and corresponding code. |
| 2 | <c:forEach var=“” items==“”> | Iterates through the elements in the "items" attribute under the name declared in "var". |
| 3 | <c:out value=“”> | Displays the content of the "value" attribute. |
| 4 | <c:set var=“” value=“” type=“” scope=“”> | Assigns the value "value" under the name "var" within the processing loop "scope" with type "type". |

#### **1.2.1.2 Commons IO Library**

The Commons IO library is a part of the Apache Commons project, providing a powerful toolkit for input/output (I/O) data processing in the Java programming language. Developed to address common challenges when working with files and data streams, Commons IO offers a range of utility methods, simplifying and optimizing I/O tasks.

This library includes various essential functions such as copying, moving, deleting files, reading/writing file contents, handling directories, and managing data streams flexibly. Commons IO helps Java programmers save time and effort when performing common I/O tasks without needing to write source code from scratch.

With its stability and popularity within the Java development community, Commons IO has become an essential tool for enhancing the performance and professionalism of Java applications in managing and processing I/O data.

***Classes for handling file paths in the Commons IO Library***

- java.nio.file.Path: This class represents a file path in the system. It provides methods for manipulating and accessing file and directory paths.

- java.nio.file.Paths: This class contains static methods for creating Path objects from strings or URIs, as well as other operations related to file paths.

#### **1.2.1.3 Commons FileUpload 2 Library**

The Commons FileUploads 2 library is an essential component of the Apache Commons project, designed to simplify the process of handling file uploads in Java applications. Built on top of the Servlet API, Commons FileUploads 2 helps developers handle input data from web browser forms, especially when files are attached.

The library provides powerful tools for parsing and processing HTTP requests containing attached data, including large files. Commons FileUploads 2 supports many features such as file size limits, efficient memory management, and support for various types of temporary storage.

The flexibility and performance of the library make handling attached files easy and secure in Java web applications. Thanks to its portability and easy integration, Commons FileUploads 2 is a preferred choice for Java projects requiring input processing from users through web forms.

***Classes for handling attached files in Commons FileUploads 2***

- FileItem: Represents a data item in an HTTP request, typically an attached file.

- DiskFileItemFactory: Responsible for creating FileItem objects, determining how FileItems will be temporarily stored before processing.

- JakartaServletDiskFileUpload: A class used to handle input data from HTTP requests sent from web browsers. It uses DiskFileItemFactory to create FileItem objects from input data, while managing the process of handling attached files. Additionally, it provides methods for processing requests and returning a list of FileItem objects.

## **1.3 Database**

### ***1.3.1 SQL Server***

Microsoft SQL Server is a relational database management system (RDBMS) developed by Microsoft. SQL Server is designed to manage and store data for enterprise and web applications. With the ability to handle both basic and large-scale data, SQL Server provides a range of powerful features.

SQL Server supports the Structured Query Language (SQL) and has the ability to optimize queries to ensure good performance. It offers features such as transactions, data security, account management, permissions management, as well as backup and restore tools.

Additionally, SQL Server integrates closely with Microsoft products and services such as Azure, Power BI, and Excel, facilitating data integration and analysis within the Microsoft environment. SQL Server is a popular solution in both the enterprise and application development communities.

***Các lệnh truy vấn của SQL Server***

|  |  |  |
| --- | --- | --- |
| STT | Lệnh truy vấn | Mô tả |
| 1 | SELECT column1, column2 FROM TableName WHERE condition | This statement selects specific columns (`column1`, `column2`) from the table named `TableName` based on the specified condition. It retrieves rows that satisfy the condition. |
| 2 | INSERT INTO table\_name (column1, column2, column3, ...)  VALUES (value1, value2, value3, ...); | inserts new rows into the specified table (`table\_name`). You specify the columns to insert data into and provide the corresponding values for each column. |
| 3 | UPDATE table\_name  SET column1 = value1, column2 = value2, ...  WHERE condition; | This statement updates existing rows in the specified table `table\_name`. It sets new values (`value1`, `value2`, ...) for the specified columns (`column1`, `column2`, ...) where the condition is met. |
| 4 | DELETE FROM table\_name WHERE condition; | This statement deletes rows from the specified table (`table\_name`) where the condition is true. It removes the entire row that satisfies the condition. |
| 5 | SELECT t1.column1, t2.column2  FROM Table1 t1  JOIN Table2 t2 ON t1.commonColumn = t2.commonColumn; | This statement performs a join operation between two tables (`Table1` and `Table2`) based on a common column (`commonColumn`). |
| 6 | CREATE TABLE table\_name (  column1 datatype,  column2 datatype,  column3 datatype,  ....  ); | Define and create a new table within a database. It specifies the name of the table and the columns it will contain, along with their data types. |

### ***1.3.2 MongoDB***

MongoDB, the most popular NoSQL database, is an open-source document-oriented database. The term ‘NoSQL’ means ‘non-relational’. It means that MongoDB isn’t based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data. This format of storage is called BSON ( similar to JSON format).

A simple MongoDB document Structure:

{

title: 'Geeksforgeeks',

by: 'Harshit Gupta',

url: 'https://www.geeksforgeeks.org',

type: 'NoSQL'

}

SQL databases store data in tabular format. This data is stored in a predefined data model which is not very much flexible for today’s real-world highly growing applications. Modern applications are more networked, social and interactive than ever. Applications are storing more and more data and are accessing it at higher rates.

Relational Database Management System(RDBMS) is not the correct choice when it comes to handling big data by the virtue of their design since they are not horizontally scalable. If the database runs on a single server, then it will reach a scaling limit. NoSQL databases are more scalable and provide superior performance. MongoDB is such a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model.

***Các lệnh truy vấn của MongoDB***

|  |  |
| --- | --- |
| **Query Command** | **Description** |
| MongoClients.create() | Establish a new client connection to the MongoDB server. |
| database.getCollection() | Get a reference to a specific collection from the database. |
| collection.insertOne(doc) | Insert a single document into a collection. |
| collection.insertMany(docs) | Insert multiple documents into a collection. |
| collection.find() | Retrieve all documents in a collection. |
| collection.find(filter) | Retrieve documents that match the given filter criteria. |
| collection.updateOne(filter, update) | Update a single document that matches the filter criteria. |
| collection.updateMany(filter, update) | Update multiple documents that match the filter criteria. |
| collection.deleteOne(filter) | Delete a single document from a collection that matches the filter criteria. |
| collection.deleteMany(filter) | Delete multiple documents from a collection that match the filter criteria. |
| collection.aggregate(pipeline) | Perform an aggregation operation using a pipeline of stages. |
| Document.parse(json) | Parse a JSON string into a Document object. |

# **2. SURVEY, ANALYSIS AND SYSTEM DESIGN**

## **2.1 Survey**

### ***2.1.1 Introduction to the Unimart Web Application***

Unimart is dedicated to enhancing the campus experience by facilitating an efficient and user-friendly platform for university students to exchange goods within the scope of their campus. Targeting the student community, Unimart aims to offer a seamless browsing and transaction experience, enabling students to buy, sell, or exchange a wide range of products, from textbooks to electronics, catering specifically to their academic and personal needs.

### ***2.1.2. Current Assessment***

***Strengths:***

* Campus-Centric Accessibility: UniMart provides easy access to a variety of goods specifically tailored for university students, accessible directly on campus.
* Convenience: The platform enables students to conduct transactions from the comfort of their dorms or anywhere on campus, saving time and effort.
* Diverse Selection: Offering a range of products relevant to campus life, UniMart caters to the diverse needs of the student population.
* Community Building: The platform encourages a sense of community among students, fostering a supportive environment for buying, selling, and exchanging goods.
* Sustainability: Encourages reuse and recycling of goods, contributing to a more sustainable campus environment.
* Time and Cost Savings: Facilitates efficient transactions that save students both time and money.

***Weaknesses:***

* Security and Privacy Concerns: Ensuring the security of transactions and the privacy of users is a critical challenge.
* Quality Assurance: Maintaining trust in the quality and condition of exchanged goods can be difficult without physical verification.

### ***2.1.3. Purpose of website***

UniMart aims to create a specialized online platform that addresses the unique marketplace needs of university students. By leveraging technology, the website intends to break down geographical barriers within campuses, offering a centralized, secure, and efficient system for students to buy, sell, and exchange goods, ultimately enhancing the overall campus experience.

### ***2.1.4. Requirement***

Functional Requirements

* User registration, login, and profile management.
* Product posting, browsing, and searching.
* Cart and order management.
* Transaction history and order management.
* Admin functionalities for platform oversight and user support.

Non-functional Requirements

* User-friendly UI.
* User-friendly interface tailored for students.
* Optimized for desktop use.
* Secure data handling and privacy protection.
* Efficient search and filter mechanisms.
* Compatibility with major browsers and devices.
* Community guidelines and support features.

## **2.2 Analysis and System Design**

### ***2.2.1. Detailed Requirements for the Website***

* Display product categories
* Display detailed information about each product
* Display best-selling, trending, popular products

For Admin:

* Manage user accounts, including the ability to suspend or delete accounts for policy violations.
* Oversee product listings for adherence to platform guidelines.
* Manage transactions and resolve disputes.
* Access an admin dashboard for insights on platform usage and trends.
* Implement and manage security measures and data protection policies.

For Customers (Students):

* View and search product listings specific to campus needs.
* Register, login, and manage personal profiles.
* Utilize a secure cart for managing desired transactions.
* Checkout and payment integration for secure transactions.
* Access transaction history and manage order details.
* Save favorite products and create cart.

For Sellers (Students):

* Post and manage their product listings.
* View and manage their sales history and active listings.

### ***2.2.2. Diagram***

**\* Usecase Diagram**

A diagram of a diagram

Description automatically generated

*Figure 2.1. General Use Case Diagram*

**Description**: Customer can register, log in, update their information, search for products, make orders, and view their orders history. Meanwhile, seller will have all the functionalities of a regular user, along with additional features for uploading products and managing products available in stock. And final, administrator will have both permission like customer and seller but mainly focus on managing platform or handling report from user.

A diagram of a network

Description automatically generated

*Figure 2.2. Specific Use Case Diagram*

**Description**: In the chart above, we gain an overview of how the Unimart operates through the customer, seller and administrator interaction steps.

**\* Sequence Diagram**

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, Hình chữ nhật

Mô tả được tạo tự động

*Figure 2.3. Login Sequence Diagram*

**Description**: When customers request to log in, they will be redirected to the login page. The system will validate whether the username and password are valid input so that to check in database. If account has in database, it will display a login successful message; otherwise, it will display a login failed message and return to the login interface.

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

*Figure 2.4. Signup Sequence Diagram*

**Description**: When users request to register as members, they will be directed to the registration page. The system will validate the entered information. If the information is valid, it will display a registration successful message. If the entered information is invalid, it will display a registration failed message.

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

*Figure 2.5. Buy Product Sequence Diagram*

**Description**: Customers add items they want to purchase to their shopping cart and it save in database as well. When user want to to buy products in cart, they require checkout. The system processes the payment and notifies the user of a successful transaction.

**\* State Diagram**

Ảnh có chứa biểu đồ, hàng, văn bản, thiết kế

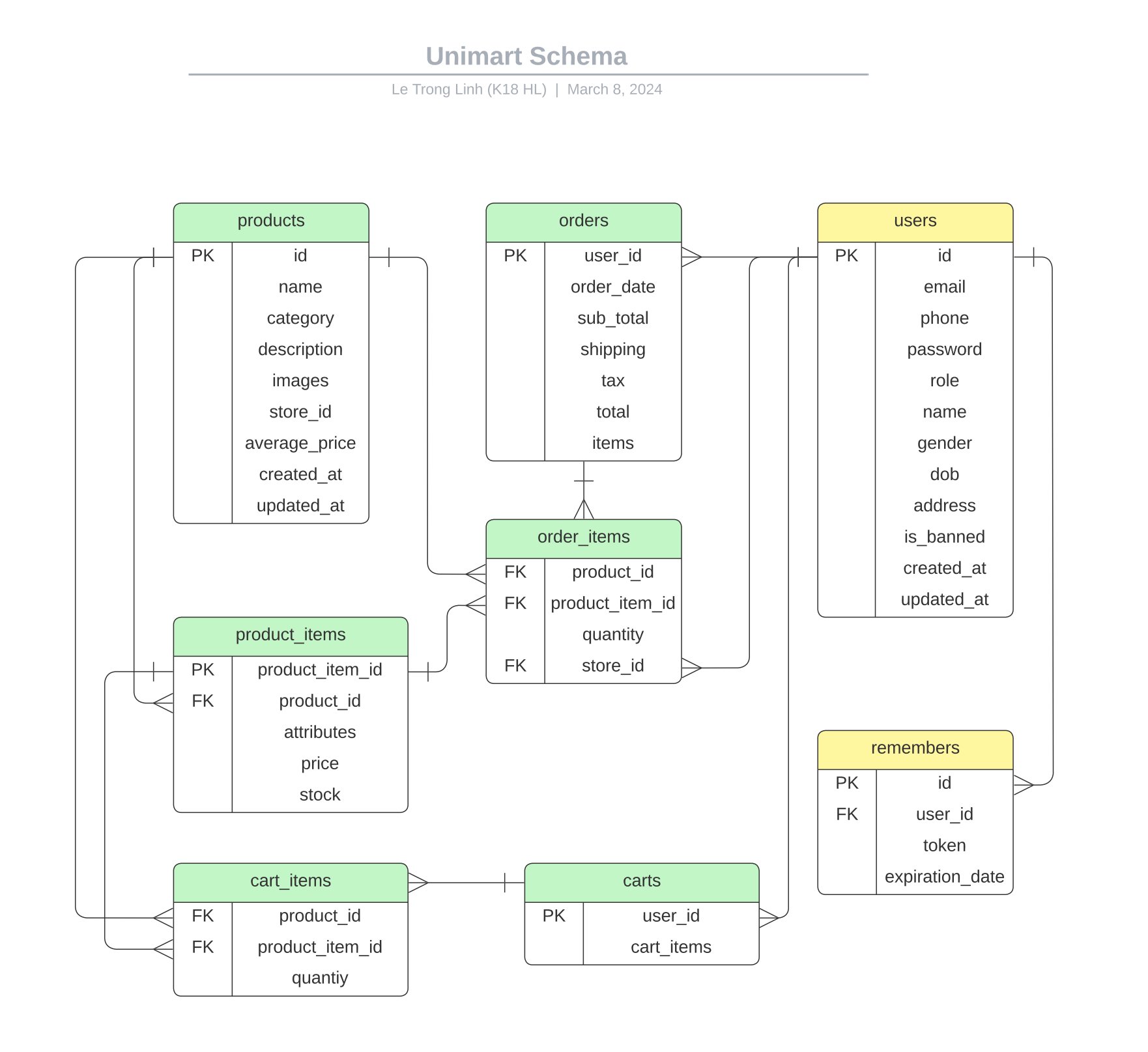
Mô tả được tạo tự động

*Figure 2.6. State Diagram for main page*

**Description**: when accessing the web, the user is in home page. If user redirect to profile page or order history page, the system check if the user is not login then redirect to login page else can access user page. According to accessing admin page, the user have to login first, if user is admin then this user can use functionality of admin page, else not.

### ***2.2.3. Database Design***

***Database Diagram***



*Figure 2.7. Database Diagram*

***Account Table (MySQL)***

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Note** |
| id | VARCHAR(100) | Primary key |
| email | NVARCHAR(50) | Unique, not null |
| phone | VARCHAR(10) | Unique, Not null |
| password | NVARCHAR(255) | Not null |
| role | ENUM('admin', ‘customer’, ‘seller’ ) | Not null |
| name | NVARCHAR(100) | Not null |
| gender | ENUM('male', 'female', 'other') |  |
| dob | DATE |  |
| address | NVARCHAR(255) |  |
| is\_banned | BIT | Not null, default: 0 |
| created\_at | DATETIME | Default: current timestamp |
| updated\_at | DATETIME | On update current timestamp |

***Remembers Table (MySQL)***

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Note** |
| id | INT | Primary key, auto-increment |
| user\_id | VARCHAR(100) | Foreign key, references id in Users table |
| token | VARCHAR(255) | Not null |
| expiration\_date | DATETIME | Not null |

***Products Collection (MongoDB)***

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Note** |
| id | String | Primary key |
| name | String | Not null |
| category | String | Not null |
| description | String |  |
| images | Array | Array of Strings (image name path) |
| store\_id | String | Reference to a Users table |
| average\_price | Double |  |
| created\_at | Date | Default: current date/time |
| updated\_at | Date | Default: current date/time |

***Product\_Items Collection (MongoDB)***

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Note** |
| Product\_item\_id | String | Primary key |
| product\_id | String | Reference to Products collection's \_id |
| attributes | Object | JSON object of key-value pairs |
| price | Double | Not null |
| stock | Integer | Not null |

***Carts Collection (MongoDB)***

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Note |
| user\_id | String | Reference to Users table |
| cart\_items | Array | Array of Object references (to Cart\_Items collection documents) |

***Cart\_Items Collection (MongoDB)***

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Note** |
| product\_id | String | Reference to Products collection's id |
| product\_item\_id | String | Reference to Product\_Items collection's id |
| quantity | Number | Not null |

***Orders Collection (MongoDB)***

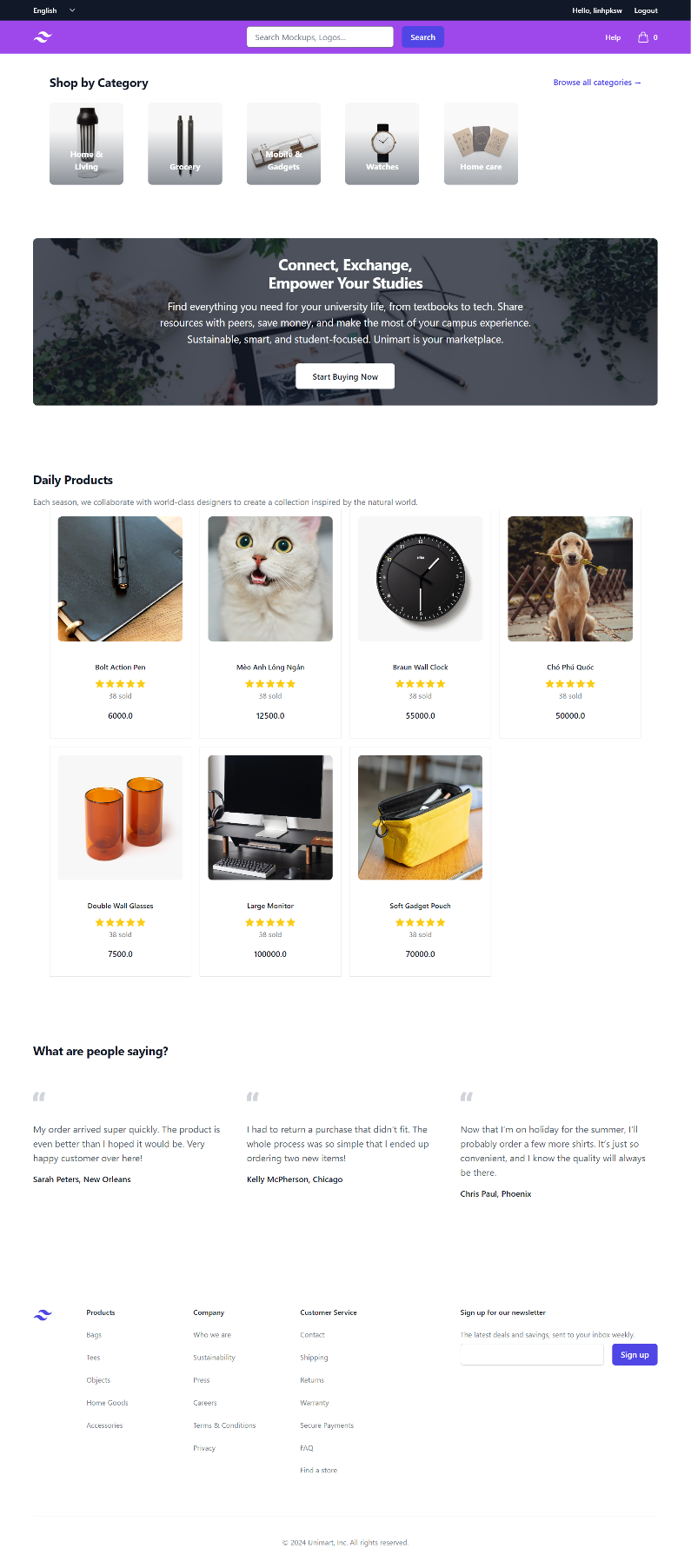
|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Note** |
| order\_id | String (6) | Unique identifier for the order |
| user\_id | String | Reference to a Users table |
| items | Array of Objects | List of order item objects |
| sub\_total | Number | Total before shipping and tax |
| shipping | Number | Shipping cost |
| tax | Number | Tax amount |
| total | Number | Total cost (sub\_total + shipping + tax) |
| order\_date | Date | The date and time when the order was placed |

***Orders Item Array (MongoDB)***

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Note** |
| product\_id | String | Reference to the Products Collection |
| product\_item\_id | String | Reference to the Product Items Collection |
| quantity | Number | Quantity of the item ordered |
| store\_id | String | Reference to the User table |

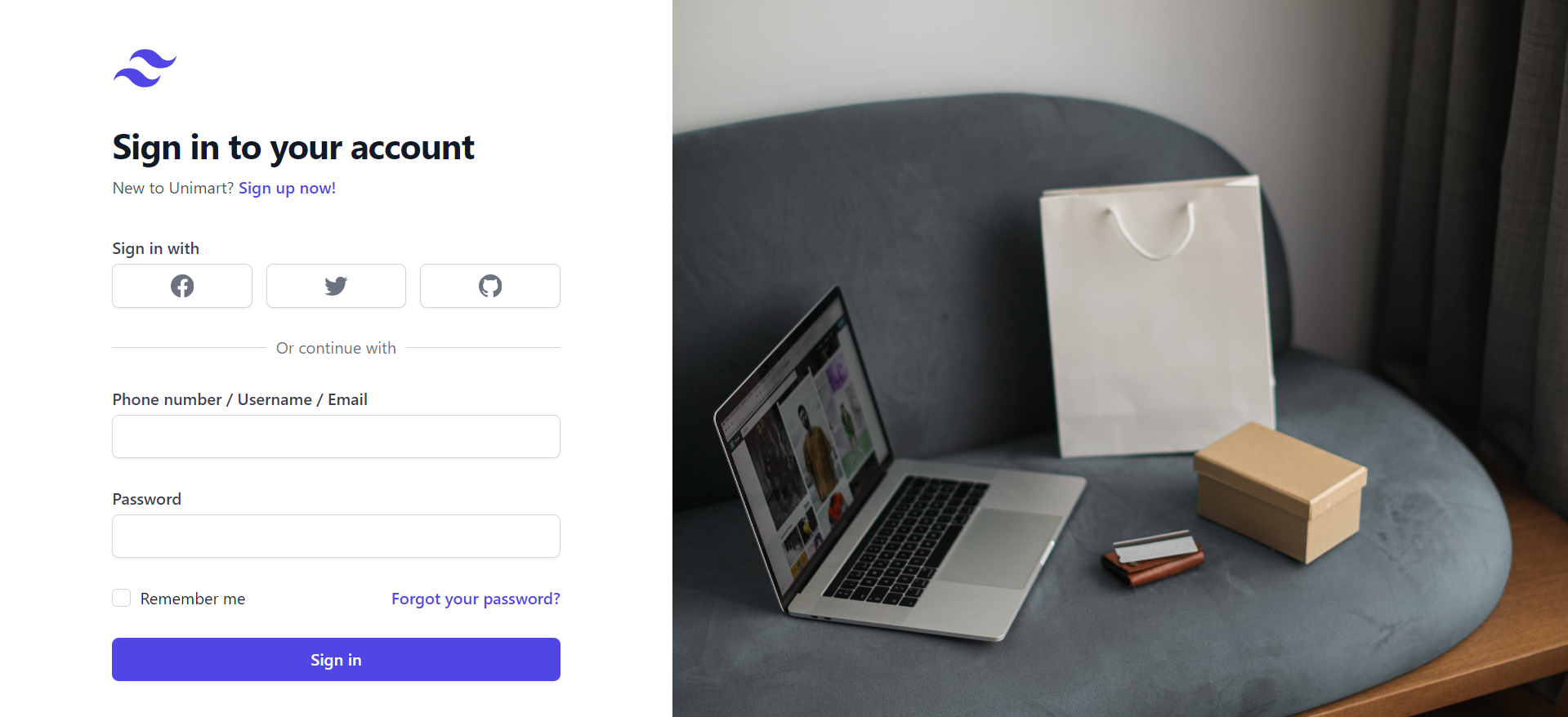
# **3. WEBSITE INTERFACE**

## ***3.1 Home Page***

****

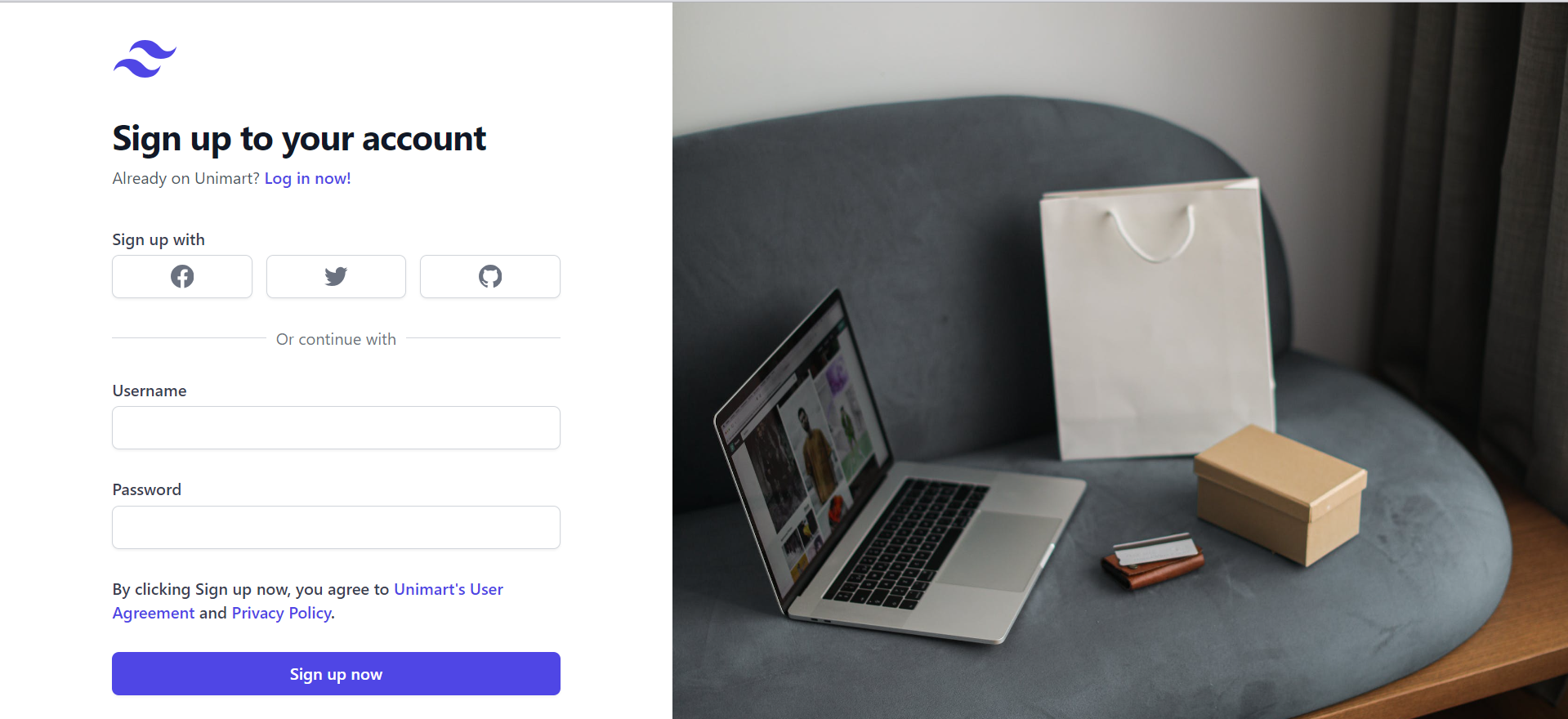
*Figure 3.1. Home page*

## ***3.2 Login Page***

****

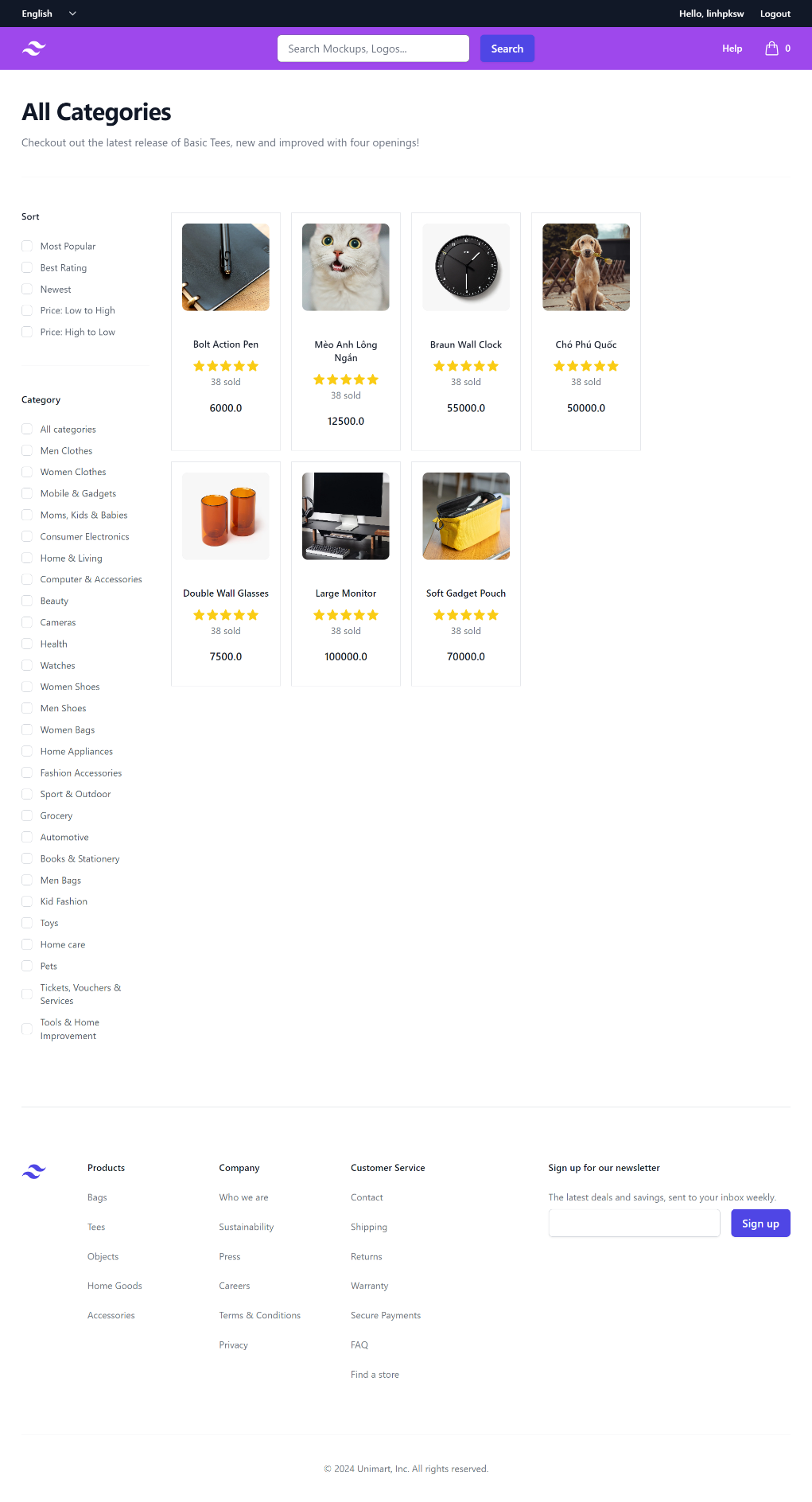
*Figure 3.2. Login page*

## ***3.3 Sign Up Page***

****

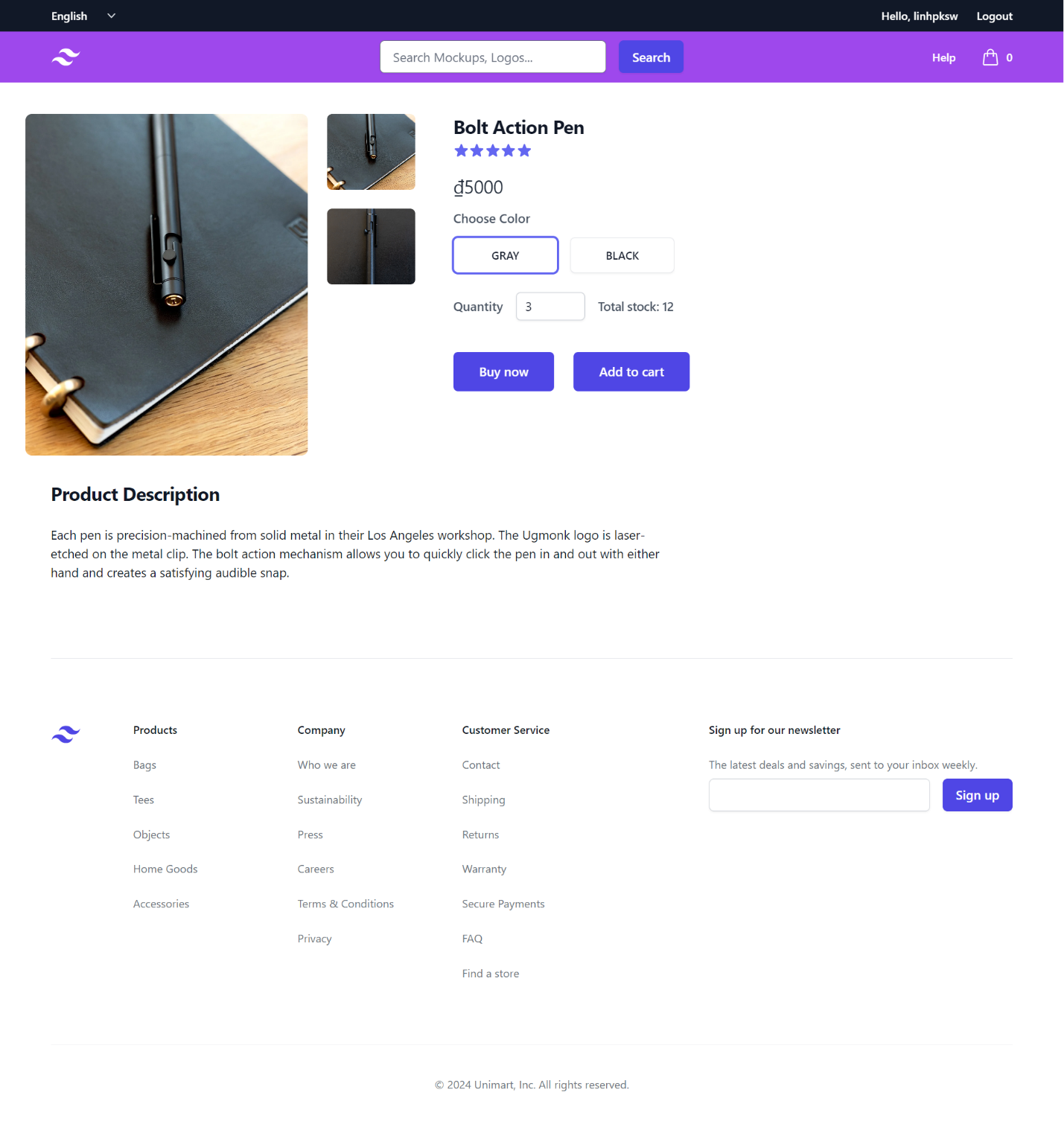
*Figure 3.3. Signup page*

## ***3.4 Collections Page***

****

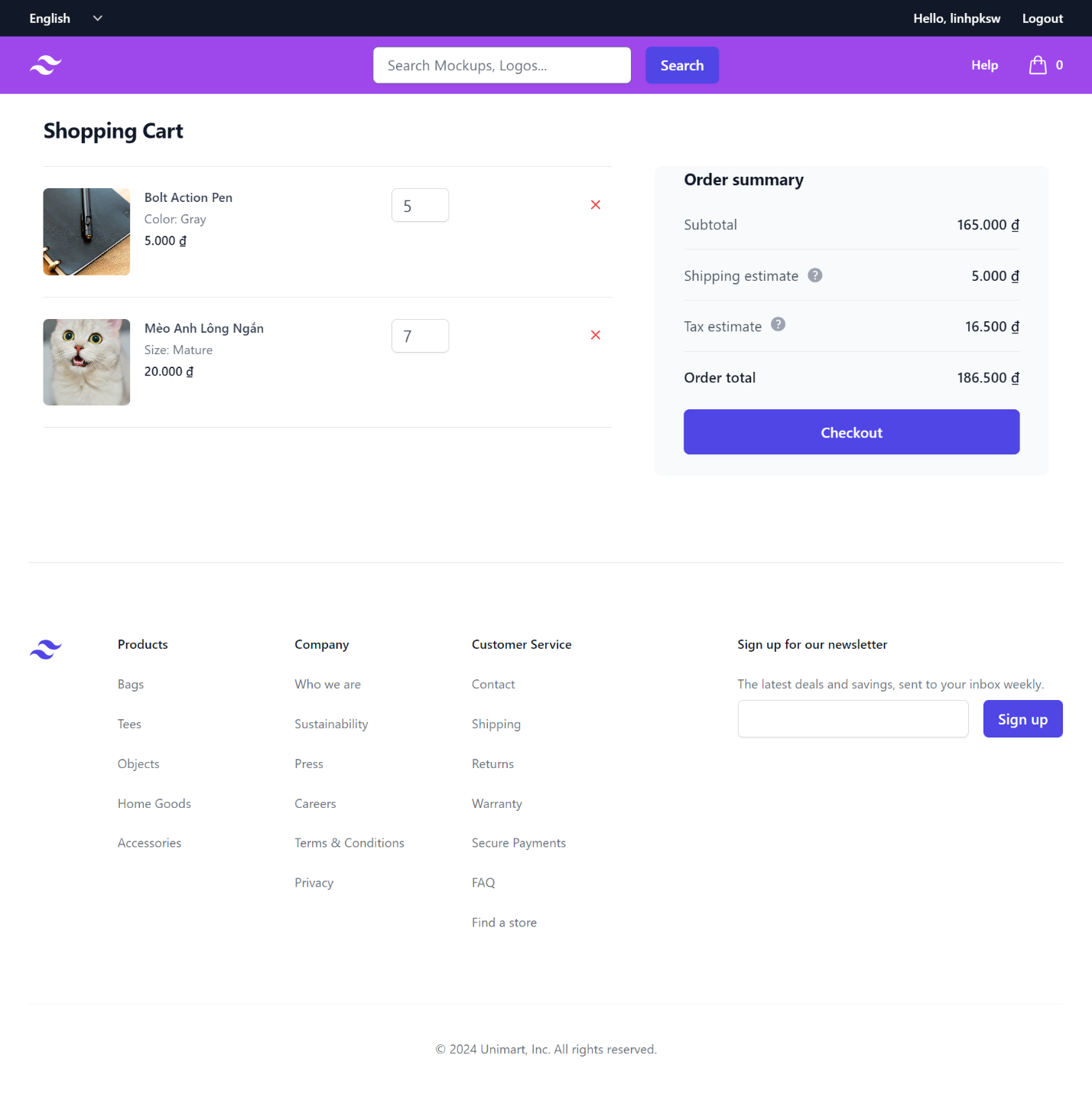
*Figure 3.4. Collection page*

## ***3.5 Product Detail Page***

****

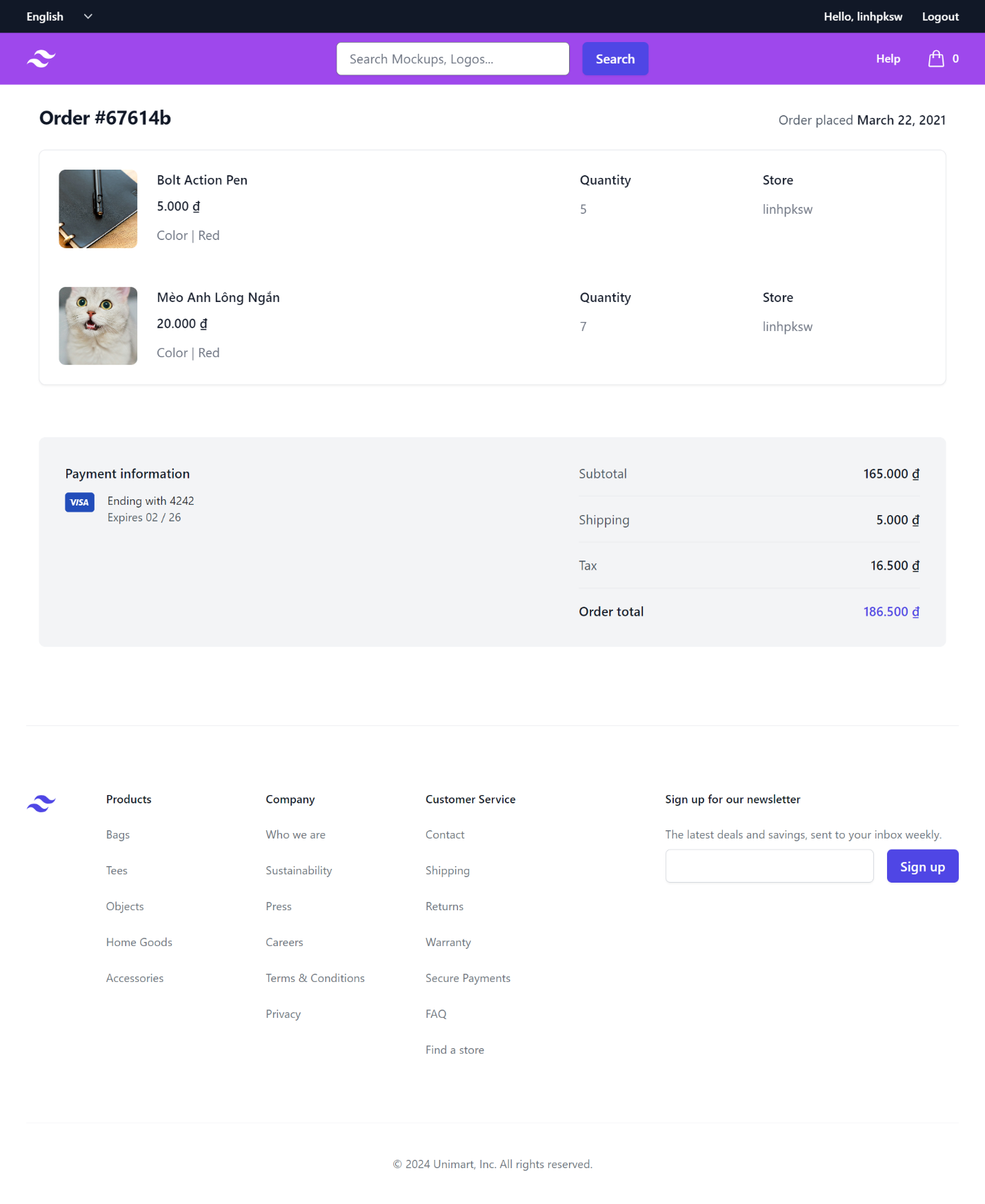
*Figure 3.5. Product detail page*

## ***3.6 Cart Page***

****

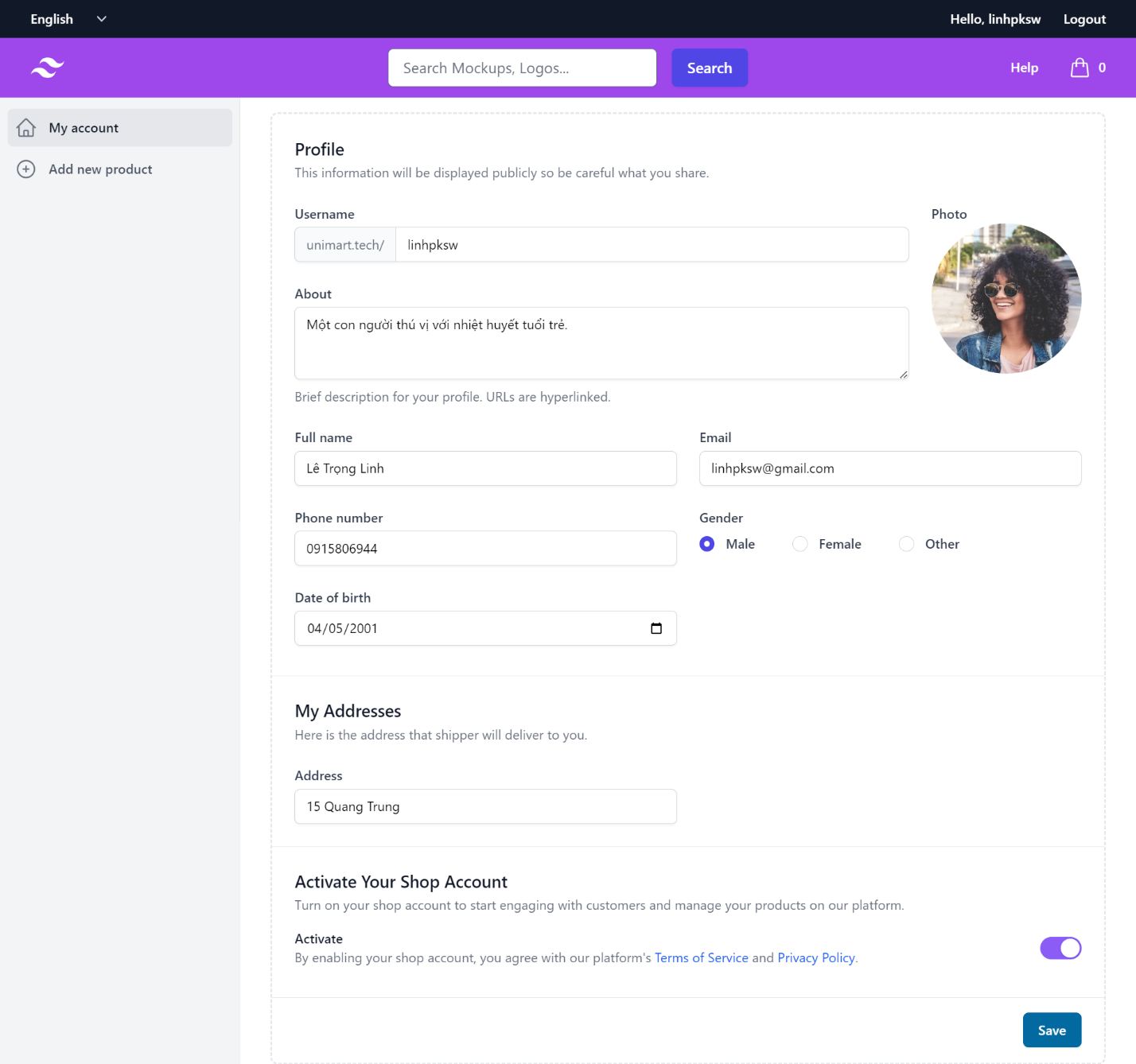
*Figure 3.6. Cart page*

## ***3.7 Order Page***

****

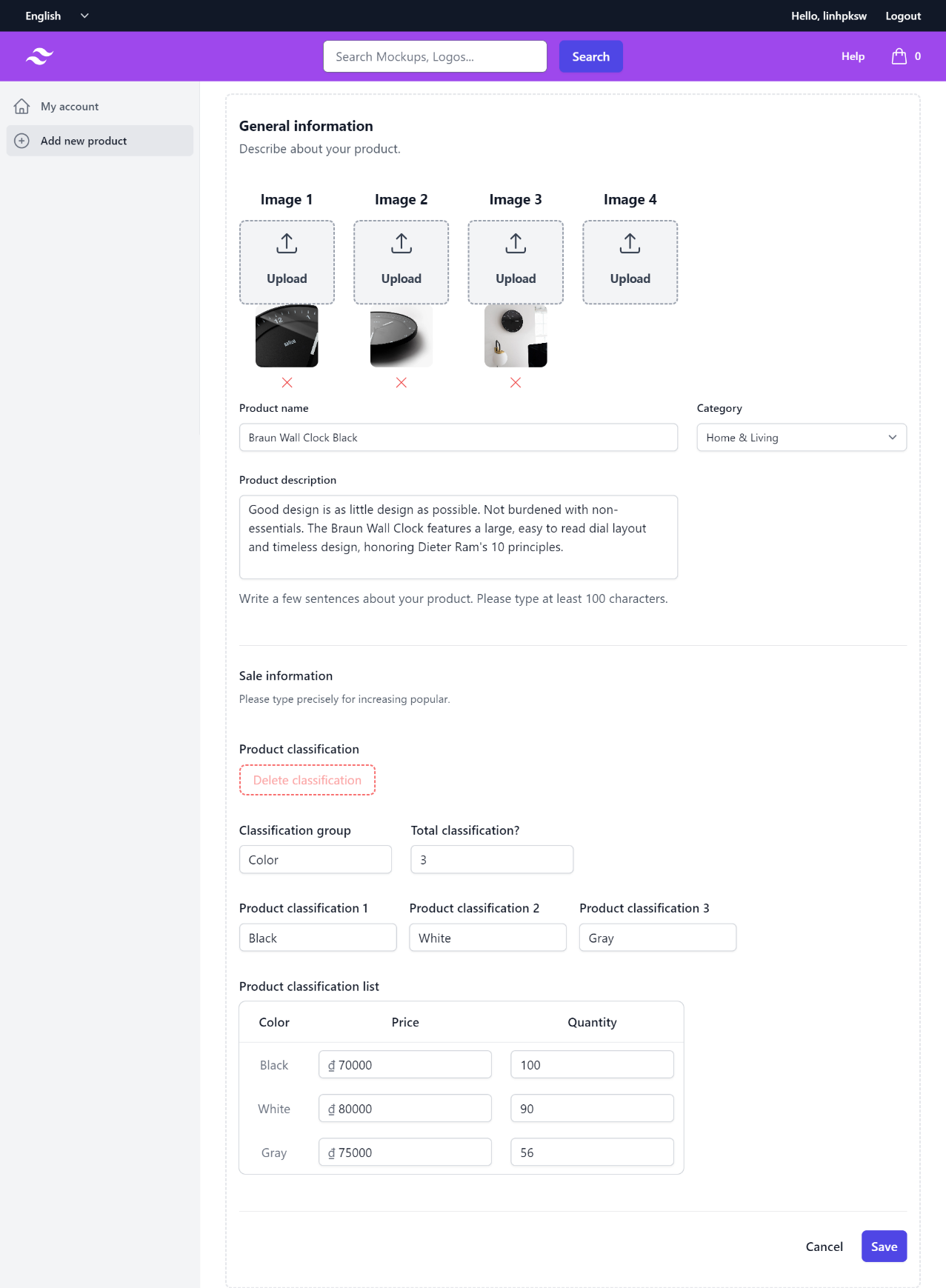
*Figure 3.7. Order page*

## ***3.8 Profile Page***

****

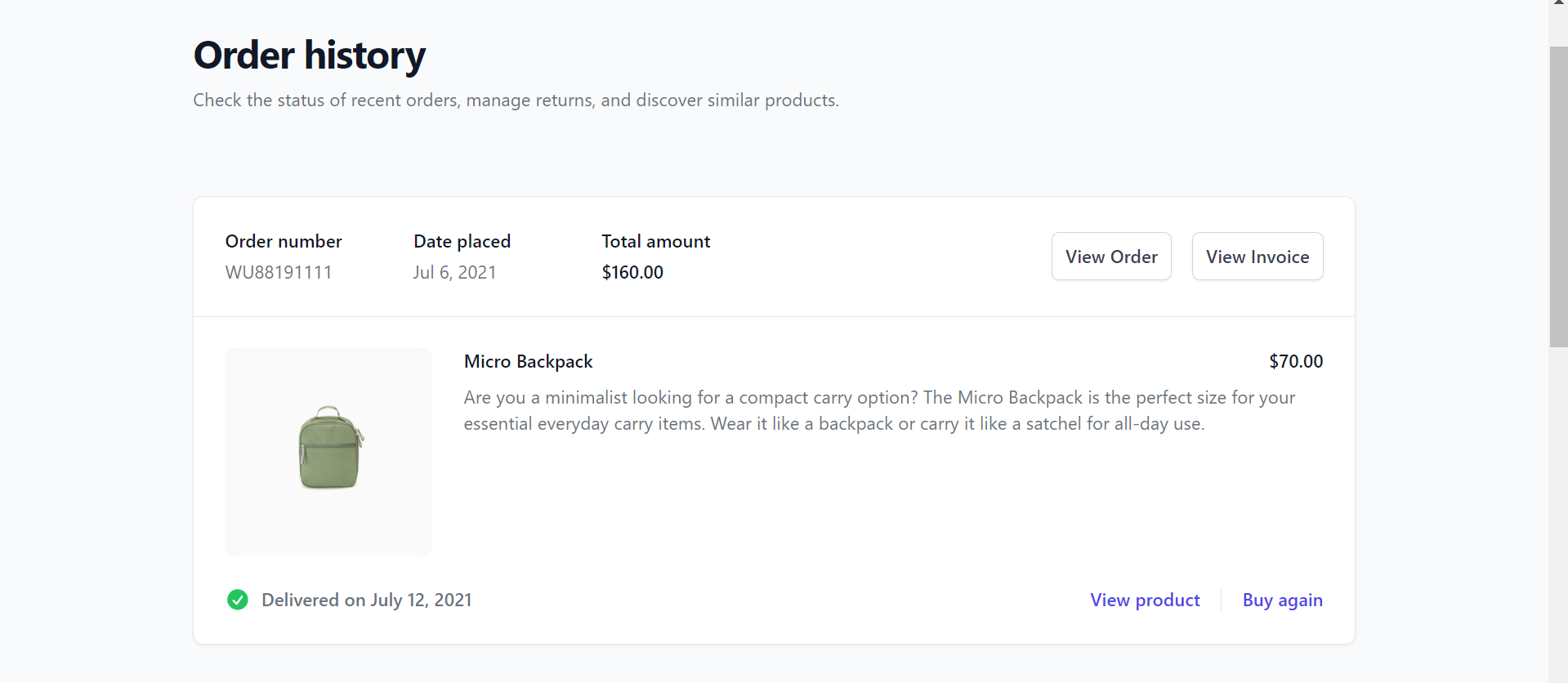
*Figure 3.8. Profile page*

## ***3.9 Add Product Page***

****

*Figure 3.9b. Add product page*

## ***3.10 Order History Page***

****

*Figure 3.10. Order history page*

# **Conclusion:**

***I. Achievements:***

• Streamlining product promotion and cart building process: The Unimart system has brought convenience and time-saving benefits to both customers and the store by digitizing the product promotion process and creating an online shopping cart.

• Simple and user-friendly interface: The system interface is designed to be simple yet professional, providing users with a good and convenient experience.

• Operational cost optimization: By operating online, Unimart has reduced costs related to physical space and labor, thereby optimizing resources and creating favorable conditions for store development.

***II. Limitations:***

• Functional deficiencies: Despite significant improvements, the system still has some functional shortcomings that may pose difficulties for users. These issues need to be addressed and resolved to improve the user experience.

• Interface shortcomings: Although designed to be simple and user-friendly, the system interface still has some areas that need improvement to ensure aesthetics and better interaction with users.

***III. Remedies and Development:***

• Optimize code to improve website performance.

• Develop the shopping cart to enable online payments.

• Expand additional application modules/plugins and deploy them on a hosting platform with a registered domain.