

# **Báo cáo dự án PRJ301**

**Tên dự án: Website Toyo**

**Trần Đình Dũng – HE186461 –  
SE1888**

## MỤC LỤC

<b>PHẦN I: CƠ SỞ LÝ THUYẾT .....</b>	<b>3</b>
<b>I. Mô hình MVC .....</b>	<b>3</b>
1. MVC là gì ? Đặc điểm của MVC .....	3
2. Các thành phần trong mô hình MVC.....	3
<b>II. Sơ lược về Servlet .....</b>	<b>3</b>
1. doGet và doPost.....	3
2. CRUD cơ bản với Servlet và Microsoft SQL Server:.....	4
3. Session.....	5
4. Cookies.....	5
5. Filter .....	5
<b>III. JSP (Jakarta Server Pages).....</b>	<b>8</b>
1. JSP và HTML .....	8
<b>IV. Framework Bootstrap dùng cho Frontend .....</b>	<b>8</b>
1. Hệ thống 12-grid system với Bootstrap 5.3 .....	8
<b>PHẦN II: KHẢO SÁT, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>	<b>10</b>
<b>I. Khảo sát.....</b>	<b>10</b>
1. Giới thiệu website ToyShop .....	10
2. Đánh giá .....	10
3. Mục đích của website .....	10
4. Giải pháp xây dựng hệ thống.....	10
<b>II. Phân tích và thiết kế hệ thống.....</b>	<b>11</b>
1. Phân tích và thiết kế hệ thống.....	11
2. Phân tích, thiết kế cơ sở dữ liệu .....	12
3. Sơ đồ Use Case.....	12
<b>PHẦN III: THIẾT KẾ WEBSITE.....</b>	<b>13</b>
1. Giao diện trang chủ.....	13
2. Giao diện đăng nhập đăng ký.....	14
3. Hiển thị sản phẩm theo danh mục.....	15
4. Hiển thị chi tiết sản phẩm .....	15
5. Giao diện giỏ hàng:.....	16
6. Giao diện thanh toán:.....	16
7. Giao diện hóa đơn:.....	17
8. Thiết kế trang giao diện quản lí.....	18
9. Hiển thị sản phẩm tìm kiếm:.....	20

III. Kỹ thuật phân trang sử dụng SQL Server và Servlet .....	20
1. Tại sao phải phân trang ? .....	20
2. Công thức phân trang và cách áp dụng: .....	20
PHẦN IV: KẾT LUẬN .....	21

## PHẦN I: CƠ SỞ LÝ THUYẾT

### I. Mô hình MVC

#### 1. MVC là gì ? Đặc điểm của MVC

MVC là viết tắt của Model-View-Controller. Cấu trúc Model-View-Controller (MVC) là một mẫu kiến trúc/mẫu thiết kế (design pattern) tách ứng dụng thành ba thành phần logic chính: Model, View và Controller. Mỗi thành phần kiến trúc được xây dựng để xử lý các khía cạnh phát triển cụ thể của một ứng dụng.

#### 2. Các thành phần trong mô hình MVC

**Model:** Model là các thành phần của ứng dụng tương ứng với tất cả logic liên quan đến miền dữ liệu (data domain), hoặc nói ngắn gọn đây là phần back-end chứa tất cả logic dữ liệu của ứng dụng. Dữ liệu ở đây có thể là dữ liệu đang được truyền giữa các thành phần View và Controller hoặc bất kỳ dữ liệu nào khác liên quan đến logic của doanh nghiệp.

Ví dụ: Giả sử bạn đang phát triển một ứng dụng mua sắm. Ở đây, Model sẽ chỉ định giỏ hàng sẽ bao gồm những dữ liệu nào — như mặt hàng, giá cả, v.v. — và những dữ liệu nào đã có sẵn trong giỏ hàng.

**View** là các thành phần hiển thị giao diện người dùng (UI) của ứng dụng. Thông thường, giao diện người dùng này được tạo từ dữ liệu Model.

Ví dụ: Trong ứng dụng mua sắm, View sẽ xác định cách hiển thị giỏ hàng cho người dùng và nhận dữ liệu từ Model để hiển thị. View sẽ bao gồm tất cả các thành phần UI như hiển thị nút bấm, danh sách thả xuống, v.v. mà người dùng cuối cùng tương tác.

**Controller:** là các thành phần xử lý tương tác của người dùng để làm việc với Model (cập nhật logic dữ liệu) hoặc/ và với View (cập nhật hiển thị giao diện người dùng).

Ví dụ: Trong ứng dụng mua sắm, ở giỏ hàng của người dùng, bạn có thể thêm các button cho phép người dùng thêm hoặc xóa các mặt hàng.

### II. Sơ lược về Servlet

#### 1. doGet và doPost

GET	POST
Dữ liệu được gửi qua phần header, giới hạn lượng dữ liệu (tối đa 2048 ký tự).	Dữ liệu được gửi qua phần body, có thể gửi lượng lớn dữ liệu.
Không an toàn vì dữ liệu hiển thị trên thanh URL.	An toàn hơn do dữ liệu không hiển thị trên URL.
Có thể đánh dấu trang.	Không thể đánh dấu trang.

<b>Idempotent:</b> các yêu cầu lặp lại không thay đổi kết quả.	<b>Non-idempotent:</b> yêu cầu lặp lại có thể thay đổi kết quả.
<b>Hiệu quả hơn và thường được dùng nhiều hơn POST.</b>	<b>Kém hiệu quả hơn và ít được sử dụng.</b>

Trong giao tiếp request-response giữa máy chủ và máy khách, hai phương thức phổ biến là:

- **GET:** Được sử dụng để yêu cầu dữ liệu từ một tài nguyên cụ thể.
- **POST:** Được sử dụng để gửi dữ liệu đã xử lý đến một tài nguyên cụ thể.

Phương thức **GET** chủ yếu để truy vấn và nhận thông tin, còn **POST** dùng khi cần gửi dữ liệu để máy chủ xử lý.

## 2. CRUD cơ bản với Servlet và Microsoft SQL Server:

Dưới đây là ví dụ cơ bản về các thao tác **CRUD (Create, Read, Update, Delete)** với **Servlet** và **MS SQL Server**.

### 2.1. Kết nối cơ sở dữ liệu:

```
protected Connection connection;
public DBContext()
{
    try {
        String user = "sa";
        String pass = "1111";
        String url = "jdbc:sqlserver://localhost\\SQLEXPRESS:1433;databaseName=ShopDB";
        Class.forName(className: "com.microsoft.sqlserver.jdbc.SQLServerDriver");
        connection = DriverManager.getConnection(url, user, password: pass);
    } catch (ClassNotFoundException | SQLException ex) {
        Logger.getLogger(name: DBContext.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}
```

### 2.2. Create(thêm dữ liệu):

```
public void createCart(int userId) {
    String query = "INSERT INTO Cart (user_id) VALUES (?)";
    try (PreparedStatement ps = connection.prepareStatement(string: query)) {
        ps.setInt(i: 1, i1: userId);
        ps.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

### 2.3. Read(đọc dữ liệu):

```

public List<User> getAll() {
    List<User> list = new ArrayList<>();
    String sql = "SELECT * FROM Users";
    try {
        PreparedStatement pre = connection.prepareStatement(string: sql);
        ResultSet rs = pre.executeQuery();
        while (rs.next()) {
            int id = rs.getInt(string: "user_id");
            String name = rs.getString(string: "username");
            String pass = rs.getString(string: "password");
            String role = rs.getString(string: "role");
            String displayName = rs.getString(string: "display_name");
            User user = new User(id, username: name, password: pass, role, displayName);
            list.add(e: user);
        }
    } catch (SQLException e) {
        System.out.println("Error fetching data: " + e.getMessage());
    }
    return list;
}

```

## 2.4. Update(cập nhật dữ liệu)

```

public boolean updateProfile(User user) {
    String sql = "UPDATE Users SET display_name = ?, password = ? WHERE username = ?";

    try (PreparedStatement stmt = connection.prepareStatement(string: sql)) {
        stmt.setString(i: 1, string: user.getDisplayName());
        stmt.setString(i: 2, string: user.getPassword());
        stmt.setString(i: 3, string: user.getUsername());
        int rowsUpdated = stmt.executeUpdate();
        return rowsUpdated > 0;
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
}

```

## 2.5. Delete(Xóa dữ liệu):

```

public void deleteUserById(int userId) {
    String sql = "DELETE FROM Users WHERE user_id = ?";
    try (PreparedStatement stmt = connection.prepareStatement(string: sql)) {

        stmt.setInt(i: 1, i1: userId);
        stmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

### 3. Session

Trong Servlet, **session** là một cơ chế giúp lưu trữ dữ liệu của người dùng trong suốt phiên làm việc giữa máy khách và máy chủ. Mỗi người dùng khi truy cập sẽ được cấp một **session ID** duy nhất, và dữ liệu có thể được lưu trữ dưới dạng các thuộc tính (attributes) trong đối tượng HttpSession. Session tồn tại trong suốt thời gian người dùng tương tác với ứng dụng (cho đến khi đóng trình duyệt hoặc hết thời gian, hay còn gọi là timeout).

- Tạo session: `HttpSession session = request.getSession();`
- Lưu thuộc tính: `session.setAttribute("key", value);`
- Lấy thuộc tính: `session.getAttribute("key");`

**Trong dự án này, Session được sử dụng để duy trì phiên đăng nhập của người dùng.**

### 4. Cookies

**Cookies trong Servlet** là cách lưu trữ dữ liệu người dùng trên trình duyệt. Máy chủ gửi một cookie đến trình duyệt thông qua phản hồi HTTP, và trình duyệt lưu trữ cookie đó để gửi lại trong các yêu cầu HTTP tiếp theo. Cookies được sử dụng để lưu trữ thông tin trạng thái như phiên đăng nhập hoặc giỏ hàng.

Tạo cookie:

```
Cookie cookie = new Cookie("key", "value");  
response.addCookie(cookie);
```

Đọc cookie:

```
Cookie[] cookies = request.getCookies();  
for (Cookie c : cookies) {  
    if (c.getName().equals("key")) {  
        String value = c.getValue();  
    }  
}
```

Các thuộc tính của Cookie

```
cookie.setMaxAge(60 * 60 * 24); // Cookie tồn tại 1 ngày
```

```
cookie.setPath("/"); // Cookie hợp lệ trên toàn bộ ứng dụng
```

Cookies giúp theo dõi thông tin như đăng nhập, sở thích người dùng, và phiên làm việc.

**Trong dự án này, Cookies được sử dụng cho phần Remember Me trong module đăng nhập (tự động điền tài khoản và mật khẩu cho người dùng đã tích vào ô Ghi nhớ đăng nhập).**

### 5. Filter

**Filter** trong Servlet là một thành phần xử lý trước hoặc sau các yêu cầu (requests) hoặc phản

hồi (responses) từ client. Nó có thể được dùng để thực hiện các tác vụ như kiểm tra xác thực,

logging mà không cần sửa đổi logic chính của Servlet.

Ví dụ về Filter:

```

public class AdminFilter implements Filter {
    @Override
    public void init(FilterConfig filterConfig) throws ServletException {

    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
        throws IOException, ServletException {

        HttpServletRequest httpRequest = (HttpServletRequest) request;
        HttpServletResponse httpResponse = (HttpServletResponse) response;

        User user = (User) httpRequest.getSession().getAttribute(string: "user");

        if (user != null && "admin".equals(anObject: user.getRole())) {
            chain.doFilter(sr: request, sr1: response);
        } else {
            httpResponse.sendRedirect(httpRequest.getContextPath() + "/home.jsp");
        }
    }

    @Override
    public void destroy() {
    }
}

```

Cấu hình trong web.xml:

```

<filter>
    <filter-name>AuthFilter</filter-name>
    <filter-class>com.example.AuthFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>AuthFilter</filter-name>
    <url-pattern>/restricted/*</url-pattern>
</filter-mapping>

```

Chức năng chính của filter:

- Kiểm tra quyền truy cập: Ví dụ như kiểm tra role để xác nhận phân quyền của tài khoản.
- Logging: Ghi lại thông tin yêu cầu, phản hồi.
- Nén nội dung: Nén dữ liệu phản hồi trước khi gửi về phía client.
- Thay đổi yêu cầu/ phản hồi: Ví dụ thay đổi header hoặc nội dung.

Filter không phải là Servlet, nhưng hoạt động tương tác với Servlet để quản lý các yêu cầu và phản hồi một cách linh hoạt.

**Trong dự án này, Filter được dùng để xác thực vai trò của người dùng, kiểm tra người dùng có quyền truy cập vào 1 trang cụ thể nào đó, kiểm tra phiên đăng nhập có đang được duy trì hay không.**

### III. JSP (Jakarta Server Pages)

#### 1. JSP và HTML

**JSP (Jakarta Server Pages)** là một công nghệ của Jakarta EE, cho phép tạo các trang web động bằng cách nhúng mã Java vào các tệp HTML. JSP giúp dễ dàng phát triển các ứng dụng web bằng cách tách phần giao diện và phần xử lý logic phía server. Các tệp JSP được biên dịch thành các **Servlet** để xử lý yêu cầu từ client.

Đặc điểm chính của JSP:

- **Tích hợp với Java:** Mã Java có thể được viết trực tiếp trong tệp JSP.
- **Các thẻ JSP:** Sử dụng các thẻ như `<%= %>` để nhúng mã Java.
- **Tái sử dụng mã:** Cho phép sử dụng **custom tags** và **JSP includes** để tái sử dụng các đoạn

mã. JSP thường được dùng để tạo các trang web động và dễ dàng duy trì.

So với **HTML**, JSP có một số lợi thế:

1. **Tạo nội dung động:** JSP có thể nhúng mã Java vào trang HTML, cho phép xử lý logic và truy vấn cơ sở dữ liệu để tạo ra nội dung động, trong khi HTML chỉ có thể hiển thị nội dung tĩnh.
2. **Tích hợp dễ dàng với Java:** JSP tích hợp trực tiếp với các công nghệ Java như **Servlet**, **JDBC**.
3. **Quản lý dễ dàng:** Tách biệt giữa giao diện (HTML) và logic (Java), giúp bảo trì dễ dàng hơn.
4. **Tái sử dụng code:** Sử dụng **taglibs** và **JSP includes** để tái sử dụng các đoạn mã.

### IV. Framework Bootstrap dùng cho Frontend

(Bài viết này chỉ đề cập đến những phần được sử dụng trong project)

#### \*Hệ thống 12-grid system với Bootstrap 5.3

Grid system trong Bootstrap 5.3 cho phép bạn xây dựng bố cục linh hoạt và đáp ứng trên nhiều kích cỡ màn hình khác nhau, sử dụng hệ thống lưới 12 cột. Bạn có thể tạo bố cục bằng các lớp CSS như `.container`, `.row`, và `.col`. Một số tính năng chính:

1. **Responsive:** Tự động thay đổi bố cục dựa trên kích cỡ màn hình (xs, sm, md, lg, xl, xxl).
2. **Gutter (khoảng cách giữa các cột):** Dùng để tạo khoảng cách giữa các cột.
3. **Breakpoint:** Cho phép kiểm soát layout theo từng kích thước cụ thể.



Dưới đây là một ví dụ:

```
<div class="container text-center">
  <div class="row">
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
  </div>
</div>
```



Ở ví dụ trên, mỗi cột chiếm 4 đơn vị ( ta sẽ có  $4 \times 3 = 12$ ) và được như trên.  
Bootstrap Grid System giúp dễ dàng tạo bố cục linh hoạt và đáp ứng cho các ứng dụng web hiện đại.

## Phần II: KHẢO SÁT, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### I. Khảo sát

#### 1. Giới thiệu về website bán đồ chơi trẻ em Toyo

- Website bán đồ chơi cho cửa hàng Toyo là nền tảng thương mại điện tử chuyên cung cấp đồ chơi cho trẻ đa dạng, chất lượng, cập nhật liên tục các loại đồ chơi mới nhất, mang đến những thú vui giải trí và đồ chơi giáo dục cho trẻ.

#### 2 Đánh giá hiện tại

##### Ưu điểm:

- **Sự đa dạng:** Cung cấp đa dạng thể loại đồ chơi cho trẻ từ đồ chơi giải trí đến đồ chơi giáo dục và trò chơi trí tuệ, v.v... đáp ứng đủ nhu cầu giải trí cho trẻ và các phụ huynh mua đồ chơi giáo dục con trẻ.
- **Tiện lợi và linh hoạt:** Người dùng có thể dễ dàng **truy cập website mọi lúc mọi nơi**, duyệt qua danh mục đồ chơi, tìm kiếm theo chủ đề, tên sản phẩm và **mua sản phẩm chỉ với vài thao tác đơn giản**.
- **Thông tin chi tiết:** Mỗi sản phẩm đều được cung cấp **thông tin đầy đủ và chính xác**, bao gồm mô tả nội dung, tóm tắt, giá bán giúp người dùng có thể đưa ra lựa chọn phù hợp.

##### Nhược điểm:

- **Hạn chế về trải nghiệm:** Khác với mua trực tiếp tại cửa hàng, khách hàng không thể trực tiếp xem chi tiết về sản phẩm nên chưa thể đưa ra cảm nhận về sản phẩm trước khi mua.

#### 3 Mục đích của website

- **Cung cấp giải pháp mua sản phẩm tiện lợi:** Mang đến cho khách hàng của cửa hàng một **phương thức mua hàng mới**, dễ dàng và linh hoạt và tiện lợi hơn so với đi đến cửa hàng mua trực tiếp.
- **Phát triển thị trường mua bán trực tuyến:** Góp phần thúc đẩy sự phát triển của thị trường mua bán trực tuyến tại Việt Nam.
- **Hỗ trợ các nhà sản xuất đồ chơi nhỏ lẻ:** Mở ra kênh bán hàng trực tuyến giúp sản phẩm tiếp cận đến nhiều khách hàng hơn.

#### 4 Giải pháp xây dựng hệ thống

##### Chức năng:

- **Trang chủ:** Hiện thị các sản phẩm bán chạy nhất, sản phẩm mới nhất, vài sản phẩm của từng danh mục, tìm kiếm nhanh, v.v...
- **Danh mục sản phẩm:** Phân loại sản phẩm theo thể loại, giá bán, cho phép người mua thêm sản phẩm vào giỏ hàng.
- **Chi tiết sản phẩm:** Cung cấp thông tin chi tiết về sản phẩm, cho phép người mua thêm sản phẩm vào giỏ hàng, đưa ra vài sản phẩm tương tự.
- **Giỏ hàng:** Quản lý các sản phẩm đã chọn, hiển thị tổng giá tiền, cho phép người dùng chỉnh sửa, thanh toán.
- **Thanh toán:** Thanh toán khi nhận hàng.
- **Hóa đơn:** Hiện thị thông tin tất cả đơn hàng của người dùng đó, admin sẽ xem được tất cả hóa đơn.
- **Tài khoản:** Cho phép người dùng đăng nhập, đăng ký, đăng xuất, ghi nhớ đăng nhập, quản lý thông tin cá nhân, xem được lịch sử mua hàng.

- **Quản lý người dùng:**Dành cho admin, quản lý tất cả người dùng có trong dữ liệu, có thể thêm sửa xóa tất cả người dùng.
- **Quản lý kho hàng:**Dành cho admin, quản lý tất cả danh mục và sản phẩm trong dữ liệu, có thể thêm sửa xóa tất cả các loại danh mục và sản phẩm.
- **Quản lý đơn hàng:**Dành cho admin, quản lý tất cả hóa đơn của tất cả người mua đã mua hàng của cửa hàng.
- **Tìm kiếm sản phẩm:**Hiện thị các sản phẩm có thể tìm được sau khi tìm kiếm bằng tên sản phẩm v.v....

#### **Phi chức năng:**

- **Giao diện:**Thiết kế giao diện hiện đại, thân thiện, dễ sử dụng, phù hợp với máy tính.
- **Hiệu suất:**Hệ thống đảm bảo ổn định, mượt mà, tốc độ tải trang nhanh chóng.
- **Bảo mật:**Đảm bảo an toàn thông tin cá nhân và tài khoản người dùng.
- **Tiếp thị:**Quảng bá website thông qua các kênh online và offline, thu hút khách hàng tiềm năng.

## **II. Phân tích và thiết kế hệ thống**

### **1 Phân tích và thiết kế hệ thống**

#### **1.1 Yêu cầu chi tiết về web**

- Hiện thị danh mục sản phẩm
- Hiện thị chi tiết về từng sản phẩm

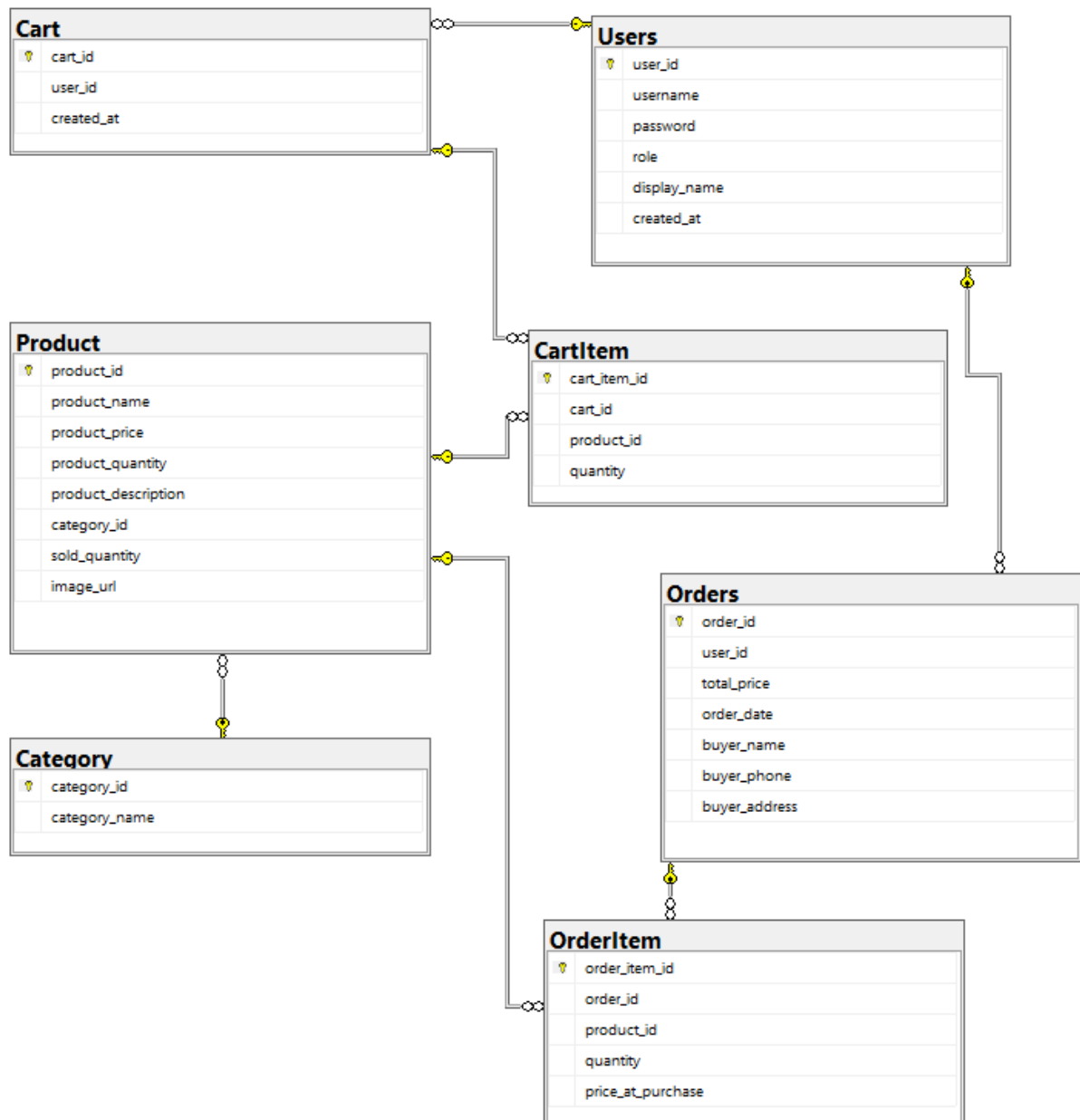
#### **\* Đối với người dùng(user)**

- Đăng nhập, đăng ký tài khoản.
- Cho phép xem thông tin sản phẩm
- Tìm kiếm sản phẩm
- Mua hàng
- Thanh toán
- Quản lý tài khoản

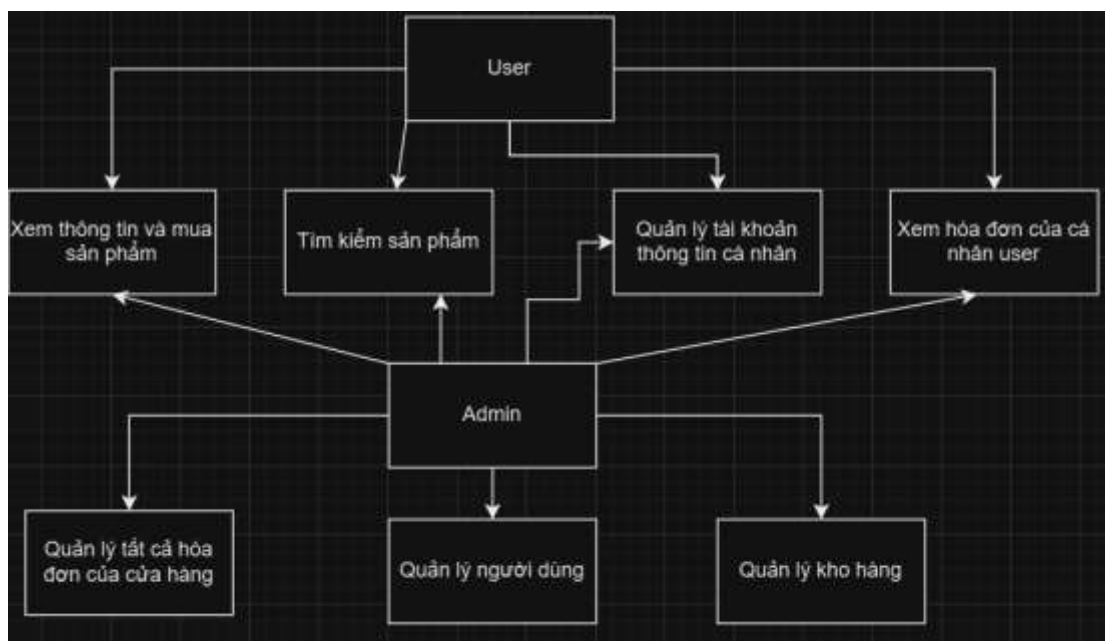
#### **\* Đối với quản trị viên(admin)**

- Thực hiện được tất cả những gì người dùng được thực hiện.
- Quản lý sản phẩm, danh mục sản phẩm: thêm sửa xóa sản phẩm, danh mục sản phẩm.
- Quản lý tài khoản người dùng: Thêm sửa xóa tất cả user có trong dữ liệu.

## 2 Sơ đồ DataBase

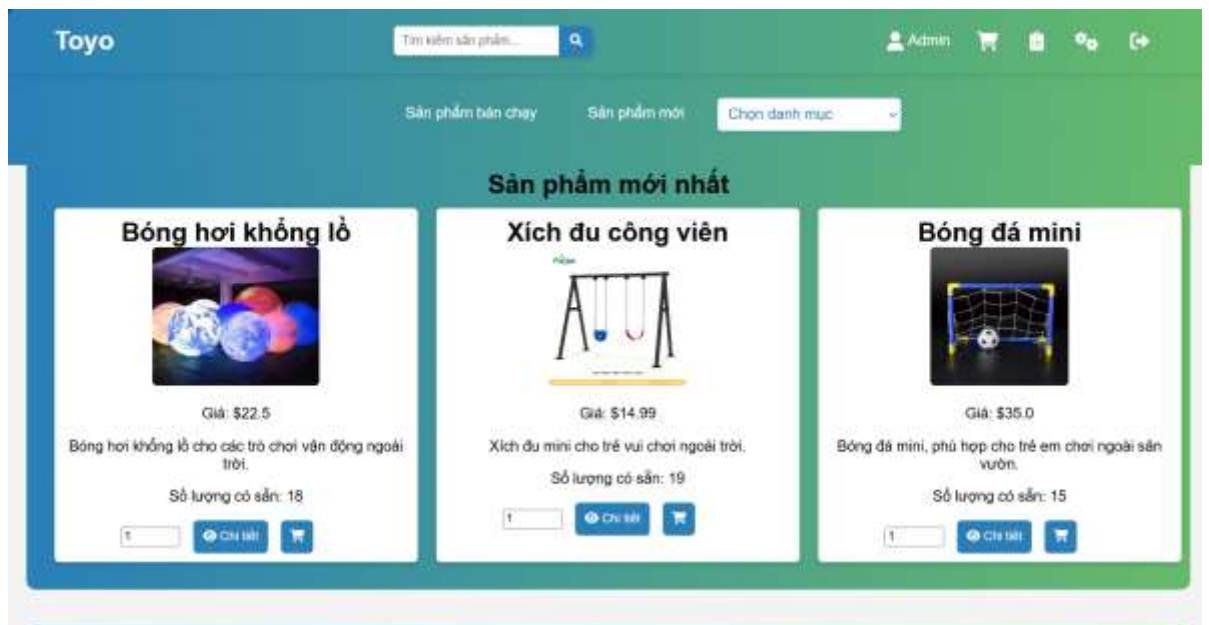
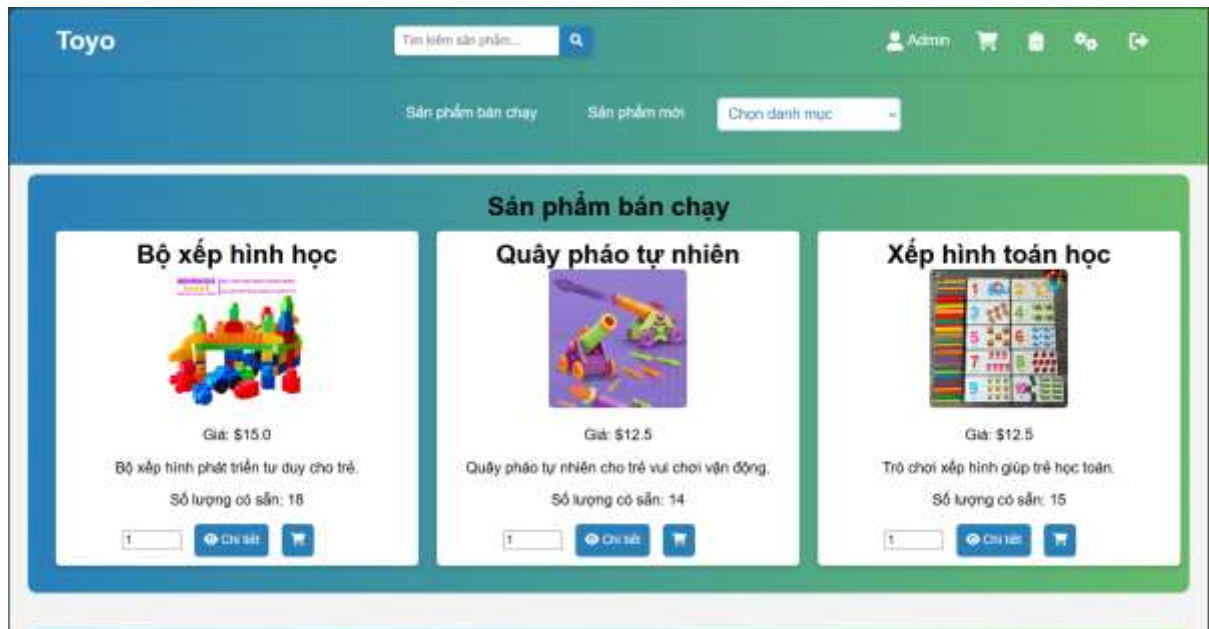


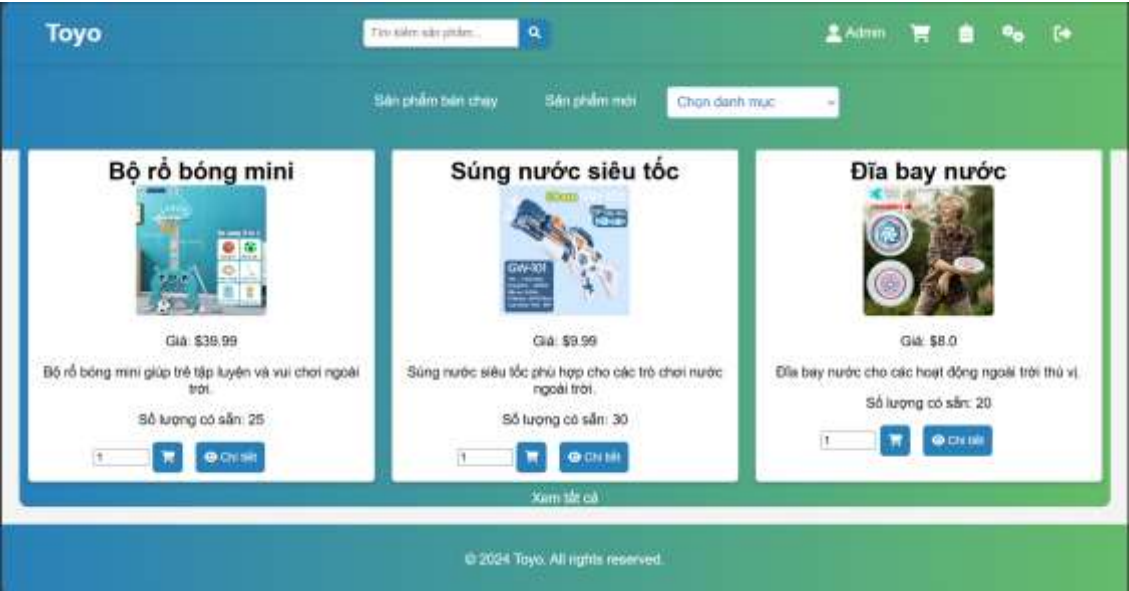
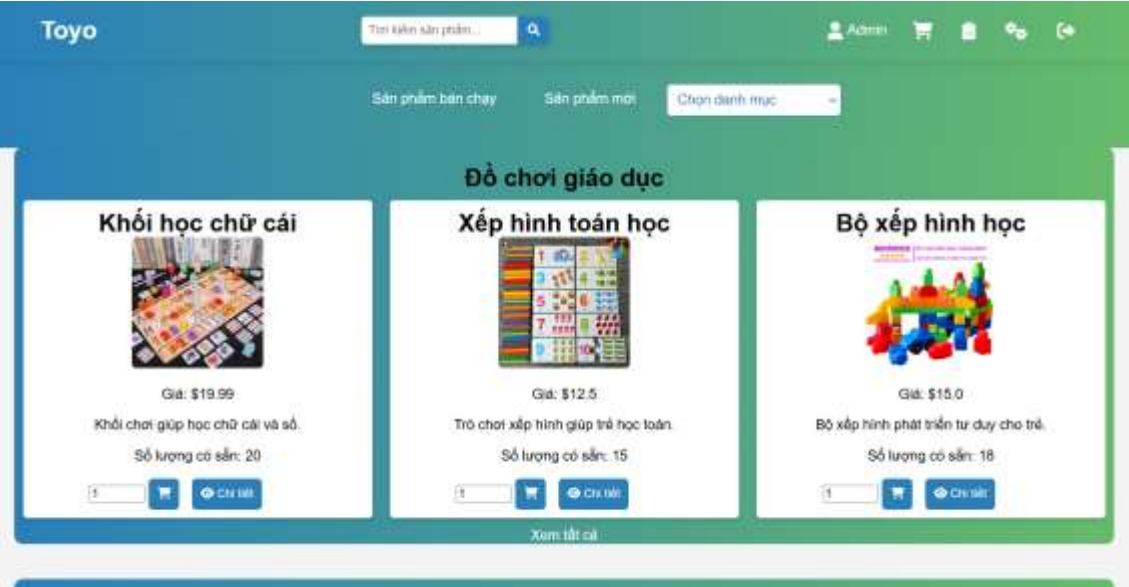
## 3 Sơ đồ Use Case



# PHẦN III: THIẾT KẾ WEBSITE

## 1 Giao diện trang chủ





## 2 Giao diện đăng nhập đăng ký.



Đăng ký

Tài khoản:

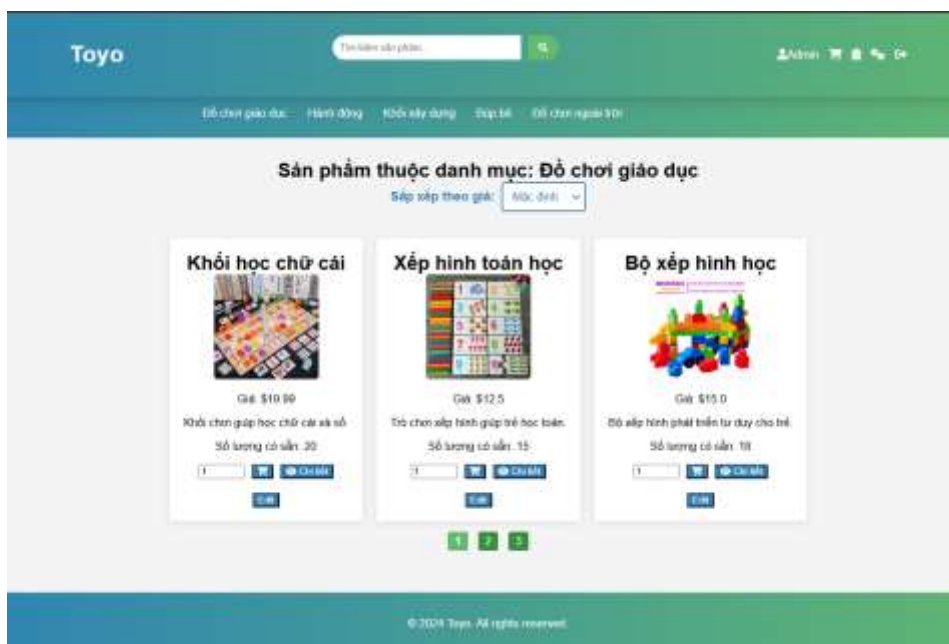
Tên hiển thị:

Mật khẩu:

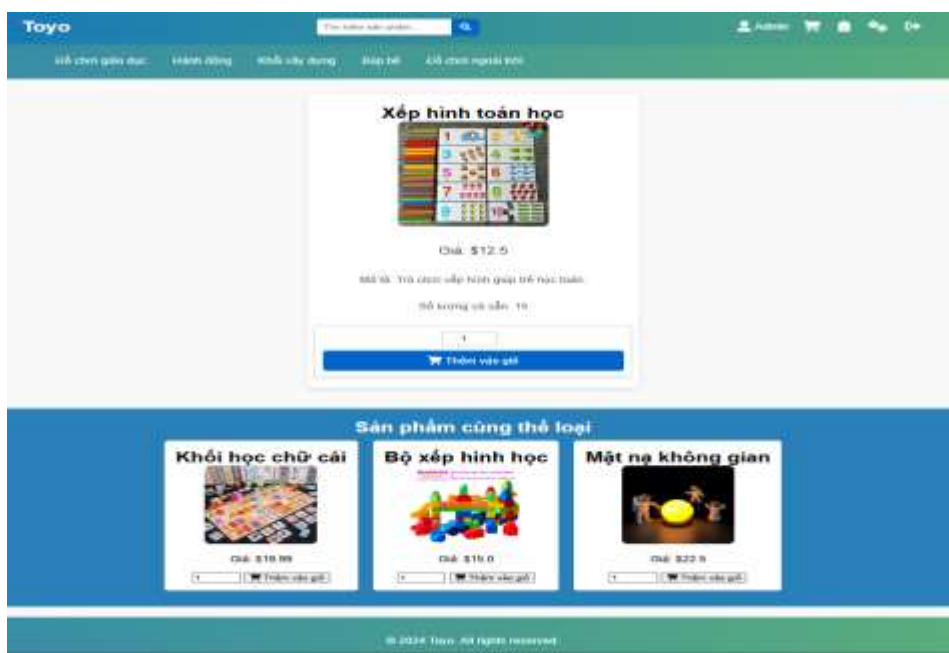
Công ty

Đã có tài khoản? Đăng nhập

### 3 Giao diện show sản phẩm theo danh mục



### 4 Giao diện show detail của sản phẩm



5 Giao diện giỏ hàng

Giỏ hàng của bạn

Tên sản phẩm	Giá (VND)	Số lượng	Tổng giá (VND)
Quầy pháo tự nhiên	12.5	<input type="text" value="1"/> <button>Cập nhật</button>	12.5 <button>Xóa</button>
Bóng hơi không lỗ	22.5	<input type="text" value="1"/> <button>Cập nhật</button>	22.5 <button>Xóa</button>
Bóng đá mini	35.0	<input type="text" value="1"/> <button>Cập nhật</button>	35.0 <button>Xóa</button>
Tổng giá trị hóa đơn: 70.0			
<button>Quay lại cửa hàng</button>		<button>Thanh toán</button>	

© 2024 Toyo. All rights reserved.

6 Giao diện thanh toán

Thanh toán

Tên sản phẩm	Giá (VND)	Số lượng	Tổng giá (VND)
Quầy pháo tự nhiên	12.5	1	12.5
Bóng hơi không lỗ	22.5	1	22.5
Bóng đá mini	35.0	1	35.0
Tổng giá trị hóa đơn:			70.0

Tên người mua:

Số điện thoại:

Địa chỉ:

Xác nhận thanh toán



## Xác nhận đơn hàng

Thanh toán thành công!

Quay lại cửa hàng

### 7 Giao diện hóa đơn

#### Danh sách đơn hàng

Đơn hàng ID	Tổng giá (VND)	Tên người mua	Điện thoại	Địa chỉ	Ngày đặt	Chi tiết
1	70.0	fdsf	0913983285	fghtfbf	2024-10-23T16:57:49.283	<a href="#">Chi tiết</a>

[Quay lại trang chủ](#)

#### Chi tiết đơn hàng

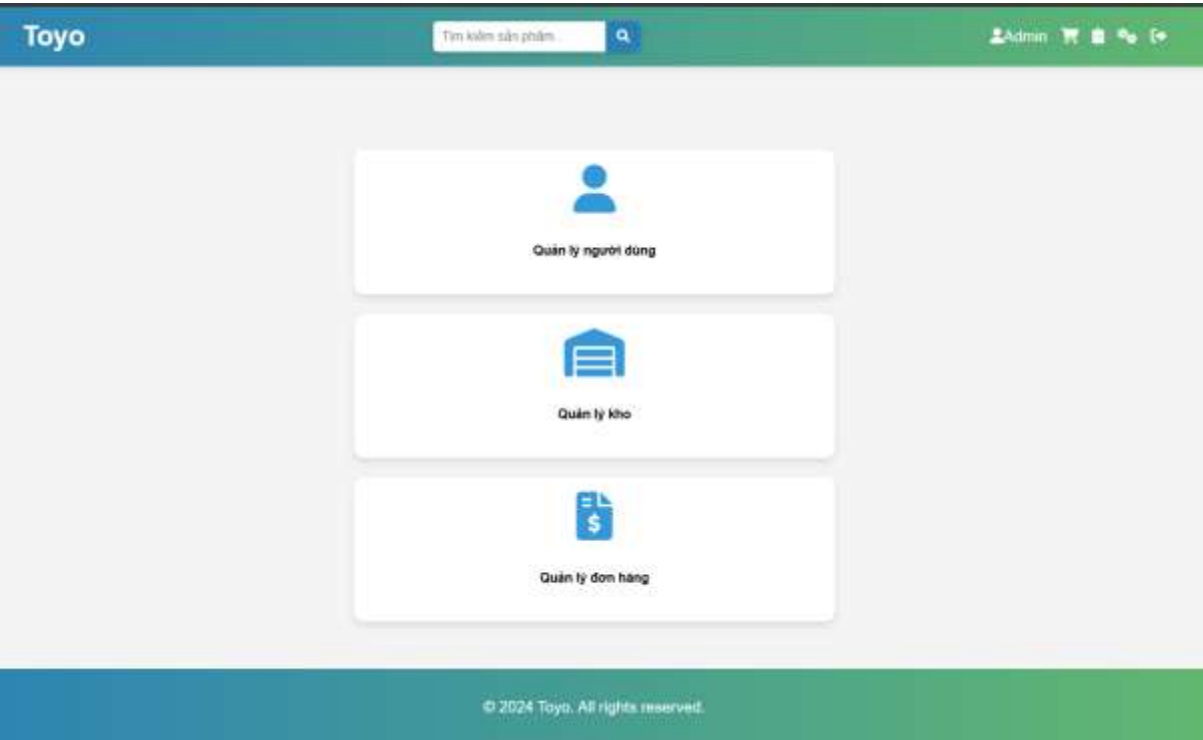
Mã đơn hàng	Tên người mua	Điện thoại	Địa chỉ	Ngày đặt	Giá trị hóa đơn
1	fdsf	0913983285	fghtfbf	2024-10-23T16:57:49.283	70.0

#### Các sản phẩm trong đơn hàng

Tên sản phẩm	Số lượng	Tổng giá
Quầy pháo tự nhiên	1	12.5
Bóng hơi khổng lồ	1	22.5
Bóng đá mini	1	35.0

[Quay lại danh sách đơn hàng](#)

8 Giao diện trang quản trị viên



\*Quản lý người dùng

Quản lý người dùng

Số lượng người dùng 3

Tài khoản	Tên hiển thị	Mật khẩu	Phân quyền	Hành động
admin	Admin	*****	admin	
dung	dung	*****	user	 
duy	duy	****	user	 


Thêm người dùng

Quay về trang quản trị

\*Quản lý kho

Quay về trang chủ

Quay về trang admin

Danh mục

Thêm danh mục

Category Name

Thêm danh mục






Categories

Tên danh mục	Số lượng sản phẩm trong danh mục	Hành động
Đồ chơi giáo dục	7	<div><div>Delete</div><div>View Products</div></div>
Hành động	7	<div><div>Delete</div><div>View Products</div></div>
Khối xây dựng	8	<div><div>Delete</div><div>View Products</div></div>
Búp bê	7	<div><div>Delete</div><div>View Products</div></div>
Đồ chơi ngoài trời	7	<div><div>Delete</div><div>View Products</div></div>

Có tất cả 36 sản phẩm

Thêm vào giỏ hàng

Tìm kiếm sản phẩm:

Product Name	Price	Quantity	Sold	Category	Image	Action
Khối học chữ cái	19.99	20	4	Đồ chơi giáo dục		<div><div>2.9k</div><div>Delete</div></div>
Xếp hình toán học	12.5	10	6	Đồ chơi giáo dục		<div><div>5.8k</div><div>Delete</div></div>
Bộ xếp hình học	15.9	18	30	Đồ chơi giáo dục		<div><div>5.8k</div><div>Delete</div></div>
Mặt nạ không gian	22.5	10	1	Đồ chơi giáo dục		<div><div>2.8k</div><div>Delete</div></div>
Hành sự vận chuyển	16.75	25	0	Đồ chơi giáo dục		<div><div>5.8k</div><div>Delete</div></div>

1

2

3

4

5

6

7

8

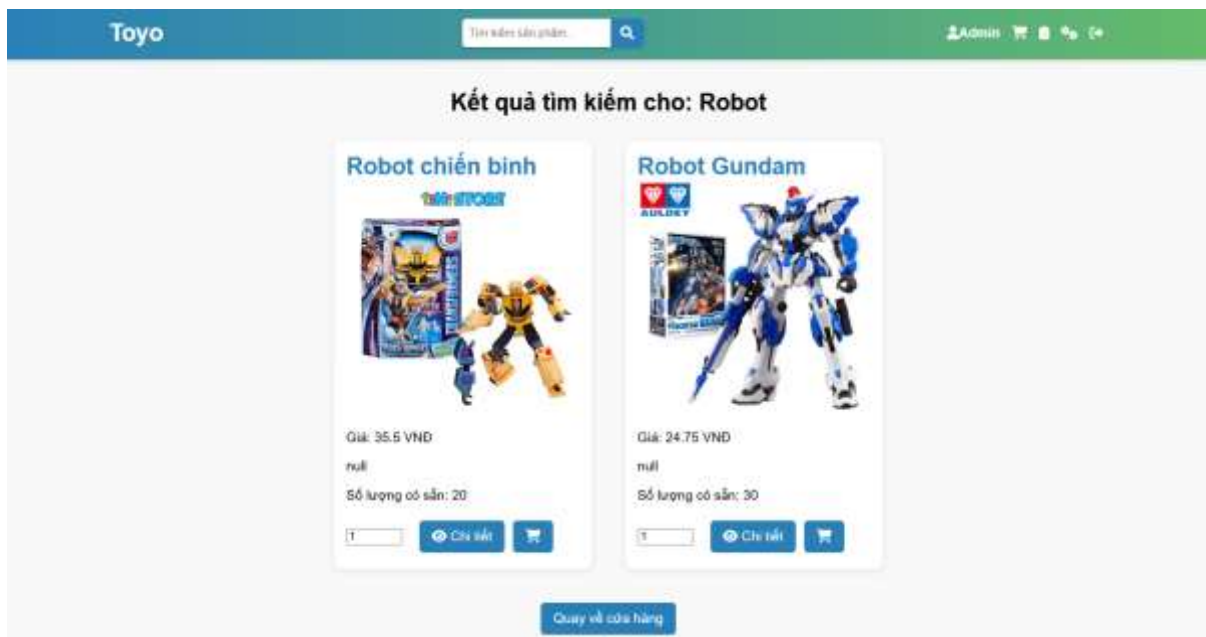
\*Quản lý hóa đơn

Danh sách Đơn hàng

Mã đơn hàng	Tên người dùng	Tổng tiền	Tên người mua	Số điện thoại người mua	Địa chỉ người mua	Thời gian mua	Chi tiết
1	admin	70.0	fsdf	0913983285	fghdtbf	2024-10-23T16:57:49.283	<div>Chi tiết</div>
2	duy	15.0	duy	0987654321	akshdajkd	2024-10-23T17:05:43.130	<div>Chi tiết</div>

Quay về trang quản trị

## 9 Tìm kiếm sản phẩm



## III. Kỹ thuật phân trang sử dụng SQL Server và Servlet

### 1. Tại sao phải phân trang ?

a. Phân trang là một phần quan trọng với bất cứ ngôn ngữ lập trình web nào. Phân trang giúp giảm thời gian tải cho website, nâng cao hiệu suất. Trong dự án này, chúng ta sẽ phân trang bằng SQL Server với câu lệnh OFFSET và FETCH NEXT ? ROWS ONLY

### 2. Công thức phân trang và cách áp dụng:

a. Các tham số cần phải có:

- recordsPerPage( số bản ghi mỗi trang, chọn một số phù hợp)
- currentPage (trang hiện tại, lấy từ tham số trang từ request.getParameter(), nếu trang load lần đầu, tự động là 1)

• offset (bỏ qua bao nhiêu bản ghi). Công thức: **offset = (page - 1) \* recordsPerPage;**

**totalRecords;**

• totalRecords( tổng số bản ghi) – truy vấn từ SQL

• totalPages (tổng số trang) **totalPages = (int) Math.ceil(totalRecords \* 1.0 / recordsPerPage);**

# KẾT LUẬN

## Kết quả đạt được:

- Hệ thống đáp ứng được phần nào nhu cầu tin học hóa khâu quảng bá sản phẩm cho cửa hàng và xây dựng giỏ hàng của khách, tiện lợi hơn, tiết kiệm thời gian của cả 2 bên mà hiệu quả không giảm.
- Hệ thống có giao diện đơn giản theo đúng xu hướng, gần gũi, thân thiện và dễ sử dụng cho cả người quản trị và khách hàng.

## Hạn chế:

Hệ thống vẫn còn nhiều thiếu sót cả về chức năng và giao diện.

## Khắc phục và phát triển:

- Hoàn thiện các chức năng còn thiếu và chưa hoàn thiện, các chức còn khá cơ bản.
- Tối ưu hóa code để cải thiện hiệu năng của website.
- Phát triển giỏ hàng để có thể thanh toán trực tuyến.
- Tăng khả năng bảo mật và chịu lỗi của website.
- Mở rộng thêm các module/plugin ứng dụng, triển khai trên host, gán tên miền.