Shop-Ease Website

E-Commerce Website

Project By:

1. **Nisha Maruti Patil**
   nishapatil19747@gmail.com

2. **Shantanu Onkareshwar Kothavale**
   shantanukothavale1@gmail.com

3. **Karan Govind Patil**
   karanpatil2300@gmail.com

ABSTRACT

Electronic Commerce, commonly known as E-commerce, revolutionizes the way we do business by harnessing the power of computer networks. It empowers individuals to access the vast resources of the Internet from the comfort of their chairs to buy or sell products. Unlike traditional commerce, where physical effort is needed to acquire goods, E-commerce simplifies our lives by reducing physical labor and saving precious time.

This revolutionary concept, which emerged in the early 1990s, has ushered in a new era in both business and technology. However, its growth has not been without hurdles, with security concerns posing a significant challenge. One of the most significant advantages of E-commerce over traditional commerce is that users can explore online shops, compare prices, and place orders for products without leaving their homes, simply by using their personal computers, tablets, etc.

Perhaps the most compelling aspect of E-commerce is its ability to bring the entire marketplace to the user's fingertips. Shoppers can now explore virtual storefronts, compare prices, and place orders, all from the comfort of their homes, armed with nothing more than a personal computer and an internet connection.

To create an e-commerce website, we must understand and use various technologies. These include multi-tiered architecture, scripting techniques for the server and client sides, and specific technologies like Angular, Spring Boot, TypeScript, and MySQL. These tools enable us to build a secure and efficient online shopping platform.

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to all the individuals that contributed to the successful development of our e-commerce website. Their support and cooperation have been invaluable throughout this journey.

In completing this project we have been fortunate to have help, support and encouragement from many people. We would like to acknowledge them for their cooperation.

First, we would like to thank Chakradhar Shinde Sir, our project advisor, for guiding us through each and every step of the process with knowledge and support. Thank you for your advice, guidance and assistance. We would like to thank him for showing immense patience and understanding throughout the project and for providing suggestions.

Finally, we would like to acknowledge the open-source community and the creators of the technologies we utilized, as their contributions have been instrumental in the development of this Shop-Ease website.

DECLARATION

We here by declaring that the project entitled, **“Shop-Ease Website”** has not been any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

TABLE OF CONTENTS

Introduction

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace. The objective of this project is to develop a general purpose e-commerce store where any product (such as books, CDs, computers, mobile phones, electronic items, and home appliances) can be bought from the comfort of home through the Internet.

An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number.

Features and Functionality of Shop-Ease website

- **Registration:**

  Registration allows new users to create accounts on the Shop-Ease website. Users provide their personal information and set up login credentials, enabling them to access the website's features and make purchases.

- **Login:**

  The login functionality is essential for r1egistered users to access their accounts securely. Users provide their login credentials (usually a username or email and password) to authenticate themselves and gain access to their personalized features.

- **Logout:**

  Logging out of the website terminates the user's current session, ensuring that their account remains secure. It prevents unauthorized access to the user's account if they are using a shared or public computer.

- **Forgot Password:**

  This feature enables users to recover their password in case they forget it.

- **Place order:**

  Users can select products they want to purchase and initiate the checkout process by placing an order. This involves confirming the items in the shopping cart, specifying the delivery address, and choosing a payment method.

- **Purchase History:**

  The purchase history feature allows users to view a record of their past orders and transactions.

- **Search products:**

  Users can search for specific products or browse through available items on the e-commerce website. The search functionality helps users find products quickly by entering keywords, categories, or filters.

- **Shopping cart:**

  The shopping cart acts as a virtual container where users can add and manage the products they intend to purchase. Users can view, modify, or remove items in the cart before proceeding to checkout.

Tools and Technologies

Front-end Technologies:

1. Angular: Angular is a popular front-end framework for building dynamic and interactive web applications. It primarily focuses on creating the user interface, managing user interactions, and displaying content to users.
2. TypeScript: TypeScript is a programming language that often serves as the foundation for developing front-end applications. It allows developers to write structured and typed code that can be compiled to JavaScript, enhancing code quality and maintainability.

Back-end Technologies:

1. Spring Boot:

   Spring Boot is a Java-based framework designed for developing robust, scalable, and efficient back-end systems.In the context of a full-stack application, Spring Boot serves as the back-end technology, responsible for handling business logic, data processing, and interactions with the database.

Database:

1. MySQL:

   MySQL is responsible for tasks such as creating and maintaining databases, executing database queries, storing user data, product information, and order details for e-commerce websites, and ensuring data integrity.

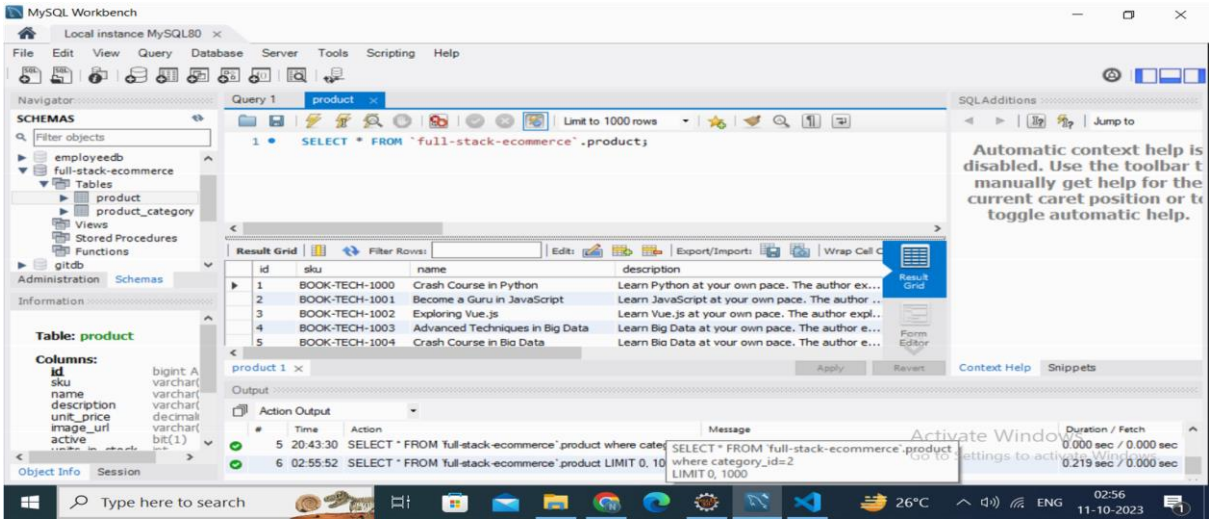
IDE:

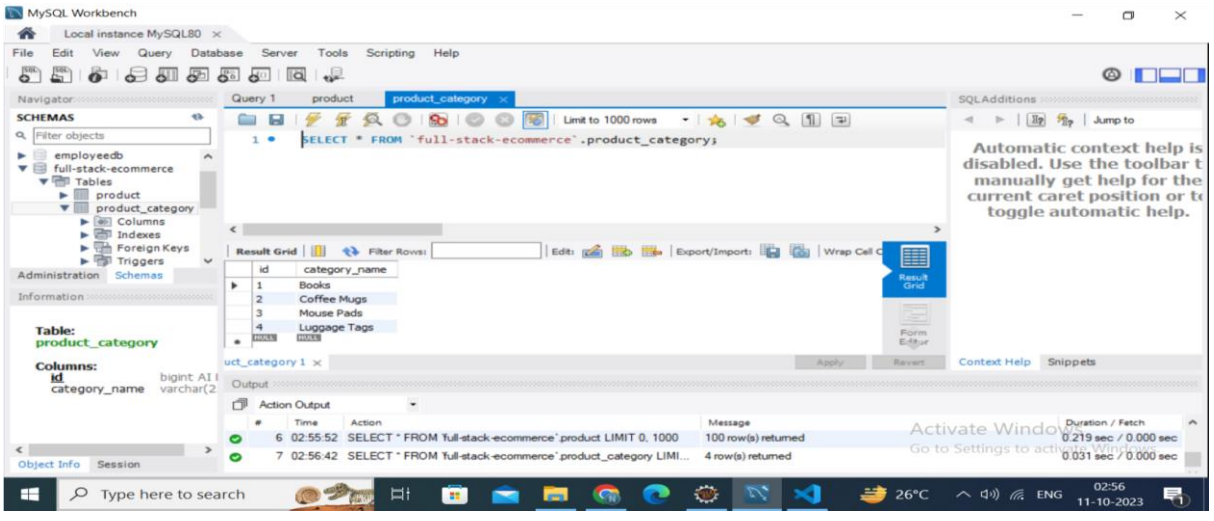
1. Eclipse
2. VS Code

Database Design

We used MySQL database for this project to store product information such as product names, descriptions, prices, images, and categories. These details are organized into database tables, making it easy to add, update, and retrieve product information. It stores user data, including user names, contact information, shipping addresses, and login credentials (encrypted passwords). This allows users to create accounts, manage their profiles, and save their shipping information for future orders.

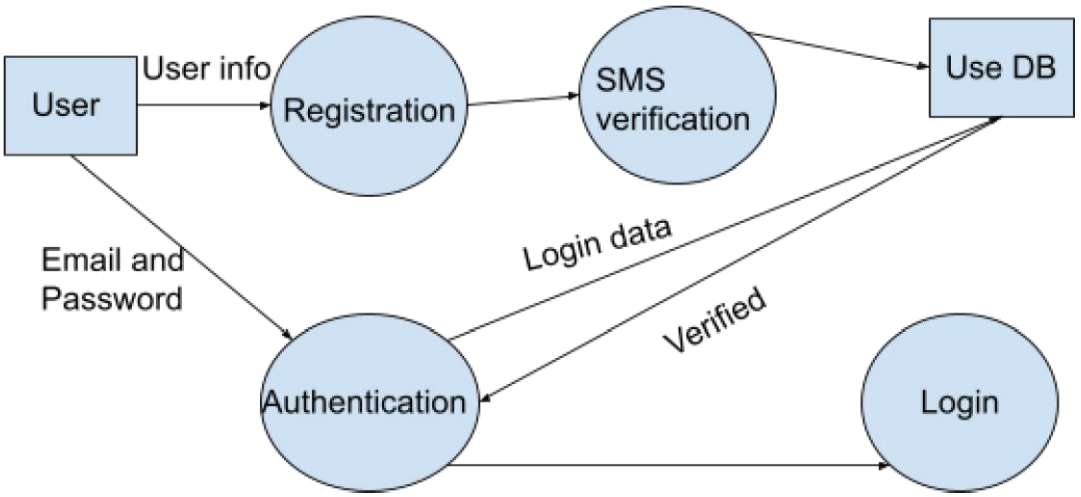
MySQL provides the necessary structure and functionality to manage and organize the vast amount of data involved in e-commerce operations. It ensures data integrity, facilitates efficient data retrieval, and contributes to the seamless functioning of the entire Shop-Ease website.



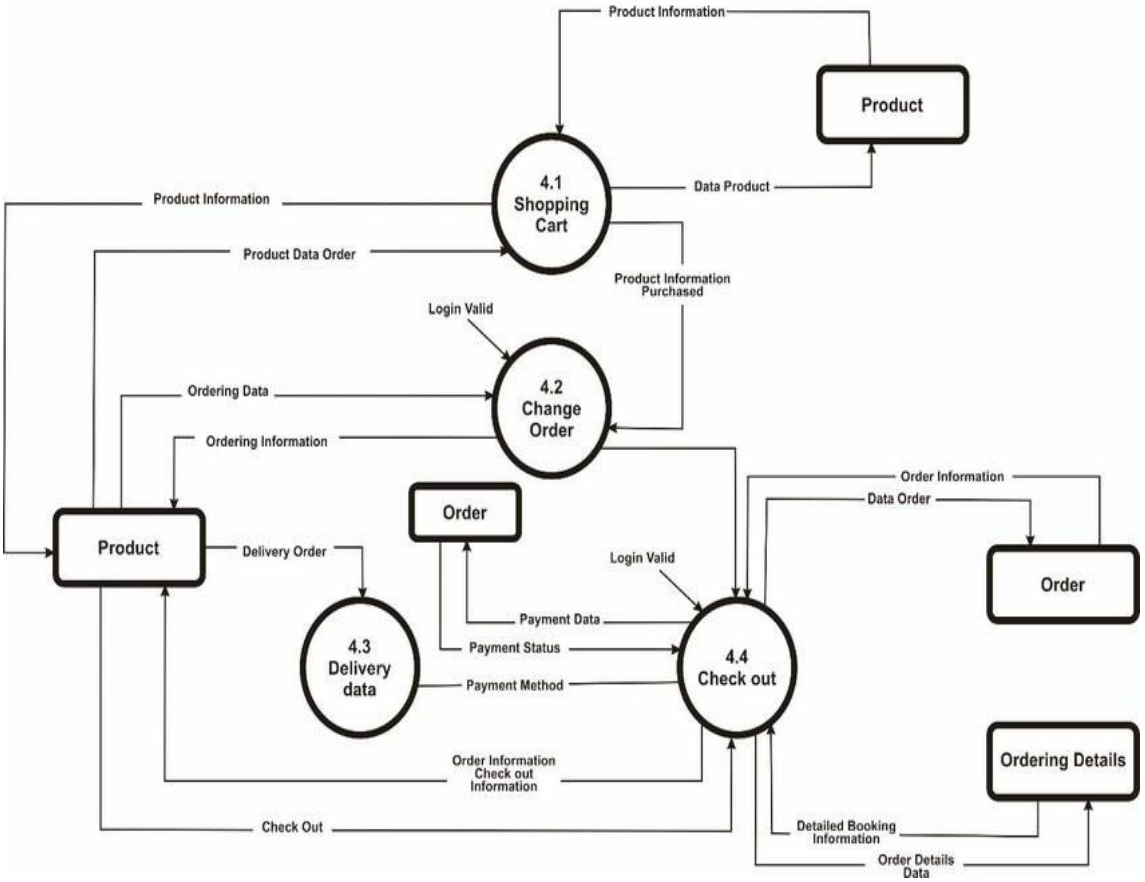


DIAGRAMS FOR SHOP-EASE WEBSITE PROJECT

1. User Registration



2. User Activity



DESIGN AND IMPLEMENTATION OF SHOP-EASE WEBSITE

Figure 29 Shopping cart

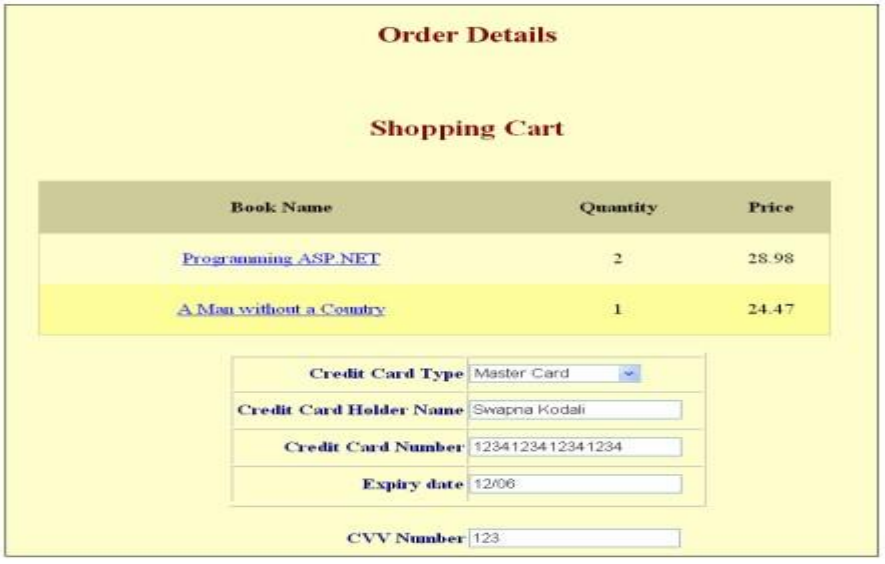


Figure 30 Order Details

Figure 32 Check out

Figure 33 Order confirmation

SOURCE CODE:

```
package com.luv2code.ecommerce; import
org.springframework.boot.SpringApplication; import
org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication public class
SpringBootEcommerceApplication { public
static void main(String[] args) {

SpringApplication.run(SpringBootEcommerceApplication.class, args);
} }


package com.luv2code.ecommerce.config; import
com.luv2code.ecommerce.entity.Product; import
com.luv2code.ecommerce.entity.ProductCategory; import
jakarta.persistence.EntityManager; import
jakarta.persistence.metamodel.EntityType; import
org.springframework.beans.factory.annotation.Autowired; import
org.springframework.context.annotation.Configuration; import
org.springframework.data.rest.core.config.RepositoryRestConfiguration; import
org.springframework.data.rest.webmvc.config.RepositoryRestConfigurer;
import org.springframework.http.HttpMethod; import
org.springframework.web.servlet.config.annotation.CorsRegistry; import
java.util.ArrayList; import java.util.List; import java.util.Set; @Configuration
public class MyDataRestConfig implements RepositoryRestConfigurer {

   private EntityManager entityManager;

   @Autowired public MyDataRestConfig(EntityManager
   theEntityManager) {

      entityManager = theEntityManager;

   }

   @Override
   public void

configureRepositoryRestConfiguration(RepositoryRestConfiguration config,

CorsRegistry cors) {

      HttpMethod[] theUnsupportedActions = {HttpMethod.PUT,

HttpMethod.POST, HttpMethod.DELETE, HttpMethod.PATCH}; // disable
      HTTP methods for Product: PUT, POST, DELETE and PATCH
      config.getExposureConfiguration()
```

```
            .forDomainType(Product.class)
            .withItemExposure((metdata, httpMethods) ->
httpMethods.disable(theUnsupportedActions))
            .withCollectionExposure((metdata, httpMethods) ->
httpMethods.disable(theUnsupportedActions));
        // disable HTTP methods for ProductCategory: PUT, POST, DELETE and
PATCH
        config.getExposureConfiguration()
            .forDomainType(ProductCategory.class)
            .withItemExposure((metdata, httpMethods) ->
httpMethods.disable(theUnsupportedActions))
            .withCollectionExposure((metdata, httpMethods) ->
httpMethods.disable(theUnsupportedActions)); // call an
internal helper method exposeIds(config);


    } private void exposeIds(RepositoryRestConfiguration
    config) {
        // expose entity ids
        //
        // - get a list of all entity classes from the entity manager
        Set<EntityType<?>> entities =
entityManager.getMetamodel().getEntities();
        // - create an array of the entity types
        List<Class> entityClasses = new ArrayList<>(); // -
        get the entity types for the entities for (EntityType
        tempEntityType : entities) {
        entityClasses.add(tempEntityType.getJavaType());
        }
        // - expose the entity ids for the array of entity/domain types
        Class[] domainTypes = entityClasses.toArray(new Class[0]);
        config.exposeIdsFor(domainTypes);
    }
}


package com.luv2code.ecommerce.dao; import
com.luv2code.ecommerce.entity.ProductCategory; import
```

```
org.springframework.data.jpa.repository.JpaRepository; import
org.springframework.data.rest.core.annotation.RepositoryRestResource;
import org.springframework.web.bind.annotation.CrossOrigin;

@CrossOrigin("http://localhost:4200")

@RepositoryRestResource(collectionResourceRel = "productCategory", path =

"product-category")

public interface ProductCategoryRepository extends

JpaRepository<ProductCategory, Long> {

}


package com.luv2code.ecommerce.dao; import
com.luv2code.ecommerce.entity.Product; import
org.springframework.data.domain.Page; import
org.springframework.data.domain.Pageable; import
org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.repository.query.Param;
import org.springframework.web.bind.annotation.CrossOrigin;

@CrossOrigin("http://localhost:4200")

public interface ProductRepository extends JpaRepository<Product, Long> {

   Page<Product> findByCategoryId(@Param("id") Long id, Pageable
pageable);

   Page<Product> findByNameContaining(@Param("name") String name,

Pageable pageable);

}
package com.luv2code.ecommerce.entity; import java.math.BigDecimal;
import java.util.Date; import com.fasterxml.jackson.annotation.JsonProperty; //
Import the annotation import org.hibernate.annotations.CreationTimestamp;
import org.hibernate.annotations.UpdateTimestamp; import
jakarta.persistence.Column; import jakarta.persistence.Entity; import
jakarta.persistence.GeneratedValue; import
jakarta.persistence.GenerationType; import jakarta.persistence.Id; import
jakarta.persistence.JoinColumn; import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table; import lombok.Data;

@Entity

@Table(name="product")

@Data
```

```
public class Product {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="id")
    private Long id;
    @ManyToOne
    @JoinColumn(name="category_id",nullable=false)
    private ProductCategory category;
    @Column(name="sku")
    @JsonProperty("sku") // Specify the JSON property name
    private String sku;
    @Column(name="name")
    @JsonProperty("name") // Specify the JSON property name
    private String name;
    @Column(name="description")
    @JsonProperty("description") // Specify the JSON property name
    private String description;
    @Column(name="unit_price")
    @JsonProperty("unit_price") // Specify the JSON property name
    private BigDecimal unitPrice;
    @Column(name="image_url")
    @JsonProperty("image_url") // Specify the JSON property name
    private String imageUrl;
    @Column(name="active")
    @JsonProperty("active") // Specify the JSON property name
    private boolean active;
    @Column(name="units_in_stock")
    @JsonProperty("units_in_stock") // Specify the JSON property
    name private int unitsInStock; @Column(name="date_created")
    @CreationTimestamp
    @JsonProperty("date_created") // Specify the JSON property name
    private Date dateCreated;
    @Column(name="last_updated")
    @UpdateTimestamp
```

```
    @JsonProperty("last_updated") // Specify the JSON property name
    private Date lastUpdated;
}
```

```
package com.luv2code.ecommerce.entity; import java.util.Set; import
com.fasterxml.jackson.annotation.JsonProperty; // Import the annotation import
jakarta.persistence.CascadeType; import jakarta.persistence.Column; import
jakarta.persistence.Entity; import jakarta.persistence.GeneratedValue; import
jakarta.persistence.GenerationType; import jakarta.persistence.Id; import
jakarta.persistence.OneToMany; import jakarta.persistence.Table; import
lombok.Getter; import lombok.Setter;

@Entity

@Table(name="product_category")

@Getter

@Setter
public class ProductCategory {

    @Id

    @GeneratedValue(strategy=GenerationType.IDENTITY)

    @Column(name="id")

    @JsonProperty("id") // Specify the JSON property name
    private Long id;

    @Column(name="category_name")

    @JsonProperty("category_name") // Specify the JSON property name
    private String categoryName;

    @OneToMany(cascade=CascadeType.ALL, mappedBy="category")
    @JsonProperty("products") // Specify the JSON property name
    private Set<Product> products;
}
```

CONCLUSION

Our e-commerce website is a big achievement. It uses technologies like Angular, Spring Boot, TypeScript, and MySQL to make shopping online easy and safe. Here's what we've learned and achieved:

- Our website is packed with cool features like making accounts, looking at products, putting them in a cart, and checking out securely.
- The MySQL database keeps all the data organized, like user info, product details, and order history. It's the storage brain of our website.
- The website looks great and works smoothly because of Angular and TypeScript. They make it easy for you to shop online from your comfy chair.

FUTURE WORK

Looking ahead, there are exciting possibilities and opportunities to make our e-commerce website even better. Here are some areas where we can focus our efforts for future development:

1. **Diversified Payment Options:** We can expand the range of payment methods to offer users more flexibility. Integrating various payment gateways, digital wallets, and cryptocurrency options can enhance the convenience and accessibility of our platform.
2. **User Experience Enhancement:** Continuously improving the user interface and experience is crucial. This involves refining the website's design, ensuring it's user-friendly, and optimizing page load times for a smoother shopping experience.
3. **Advanced Recommendation Systems:** Implementing advanced recommendation algorithms can help users discover products they're likely to be interested in, increasing user engagement and sales. Personalized product suggestions can significantly enhance the shopping experience.
4. **Security Strengthening:** E-commerce websites are prime targets for cyber threats. Implementing advanced encryption methods and proactive security measures is paramount.

In the future, we'll keep making the website better. The road ahead is full of opportunities to make online shopping even more awesome.