



Clarkton Corporation

Design Specification Document

Table of Contents

1	Introduction	1
2	Project Goals	1
3	Target Audience	1
4	User Personas.....	2
5	Functional Requirements	3
5.1	User Authentication and Authorization	3
5.2	Comprehensive Product Catalog.....	4
5.3	User Friendly Search and Filtering	4
5.4	Intuitive Shopping cart and Checkout Process.....	4
5.5	Secure Payment Gateway Integration.....	4
6	Use Case Diagram	5
7	UML Diagram	6
8	Architecture Design.....	6
8.1	Bussiness Logic Layer	8
8.2	Data Layer	9
8.3	Deployment Layer	11
9	Database Design.....	12
10	Scalability	15
11	Security	16
12	API Documentation	18
13	User Interfaces	20
14	Developer Documentation.....	31

1. Introduction

The Clarkton Corporation website is an e-commerce platform designed for buying and selling auto parts in Worcester, Massachusetts. The website is built using the MERN stack (MongoDB, Express, React, and Node.js) and Bootstrap 5, ensuring a robust and scalable infrastructure.

2. Project Goals

The primary objectives of the Clarkton Corporation website are to:

- Establish a user-friendly and responsive e-commerce platform that caters to a diverse range of users, including DIY auto enthusiasts, auto repair shops, and businesses involved in auto part resales.
- Implement a secure and reliable payment gateway integration, enabling users to seamlessly process online transactions with confidence.
- Create a comprehensive catalog of auto parts, encompassing a wide variety of products with detailed descriptions, high-quality images, and comprehensive specifications.
- Offer a range of shipping options and delivery methods to accommodate the diverse needs of customers, ensuring timely and efficient delivery of purchased auto parts.
- Implement a responsive customer support system to effectively address customer inquiries, concerns, and potential issues promptly and professionally.

3. Target Audience

The target audience for the Clarkton Corporation website encompasses three primary groups:

- **DIY Auto Enthusiasts:** Individuals with a passion for automotive maintenance and repairs, seeking easy access to parts, detailed product information, and competitive pricing.
- **Auto Repair Shop Owners:** Experienced automotive professionals managing auto repair businesses, requiring a reliable parts supply, bulk discounts, and an efficient ordering process.
- **Auto Parts Resellers:** Individuals or businesses engaged in the resale of auto parts, seeking access to a wide range of parts, wholesale pricing, and drop shipping options to streamline their operations.

4. User Personas

To effectively cater to the needs of our diverse target audience, we have developed user personas that represent the key demographics and usage patterns:

Persona 1: DIY Auto Enthusiast

- Age: 25-55 years old
- Gender: Predominantly male
- Experience Level: Intermediate to advanced automotive knowledge
- Needs: Easy-to-navigate website, detailed product information, competitive pricing, and straightforward purchasing process

Persona 2: Auto Repair Shop Owner

- Age: 35-65 years old
- Gender: Predominantly male
- Experience Level: Extensive automotive knowledge and experience
- Needs: Reliable parts supply, bulk discounts, efficient ordering process, and integration with existing inventory management systems

Persona 3: Auto Parts Reseller

- Age: 30-60 years old
- Gender: Male or female
- Experience Level: Business acumen and familiarity with the auto parts industry
- Needs: Access to a wide range of parts, wholesale pricing, drop shipping options, and streamlined order processing to enhance customer satisfaction

5. Functional Requirements

To achieve the project goals and cater to the needs of our target audience, the Clarkton Corporation website will implement the following functional requirements:

User Authentication and Authorization:

- Users should be able to create secure accounts and log in seamlessly.
- Users should be able to manage their account information, including profile details and shipping addresses.
- Implement role-based access control (RBAC) to restrict access to sensitive information based on user roles.

Comprehensive Product Catalog:

- The website should feature a comprehensive catalog of auto parts, encompassing a wide range of products with detailed descriptions.
- Implement a robust search function to enable users to easily find products using relevant keywords, categories, and manufacturers.

- Provide high-quality images and comprehensive specifications for each product to assist users in making informed purchasing decisions.

User-Friendly Search and Filtering:

- Implement advanced product filtering options to allow users to narrow down their search results based on specific criteria, such as price, brand, compatibility, and other relevant attributes.
- Provide intuitive search filters that are easily accessible and customizable to cater to the diverse needs of users.

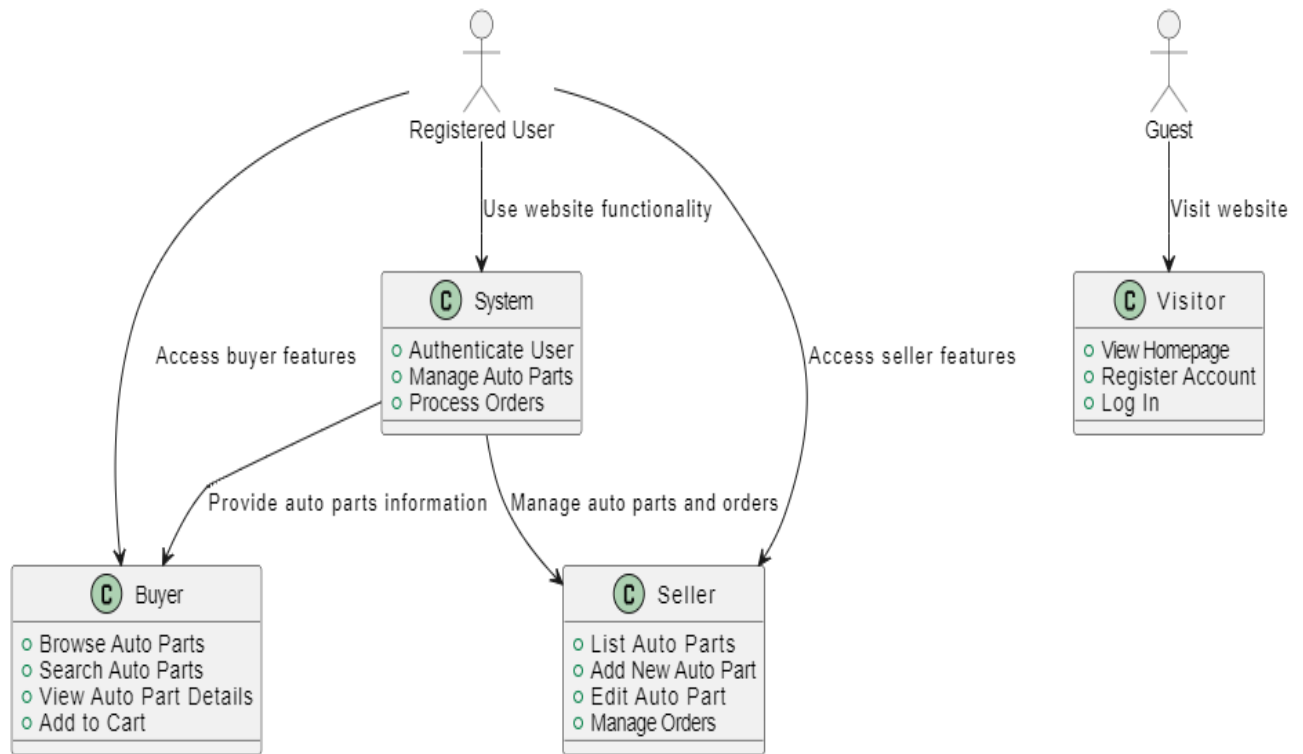
Intuitive Shopping Cart and Checkout Process:

- Implement a user-friendly shopping cart system that allows users to add and remove items seamlessly.
- Provide a clear and concise checkout process that guides users through each step of the purchasing process.
- Offer a variety of payment options, including credit cards, debit cards, and digital wallets, to accommodate user preferences.

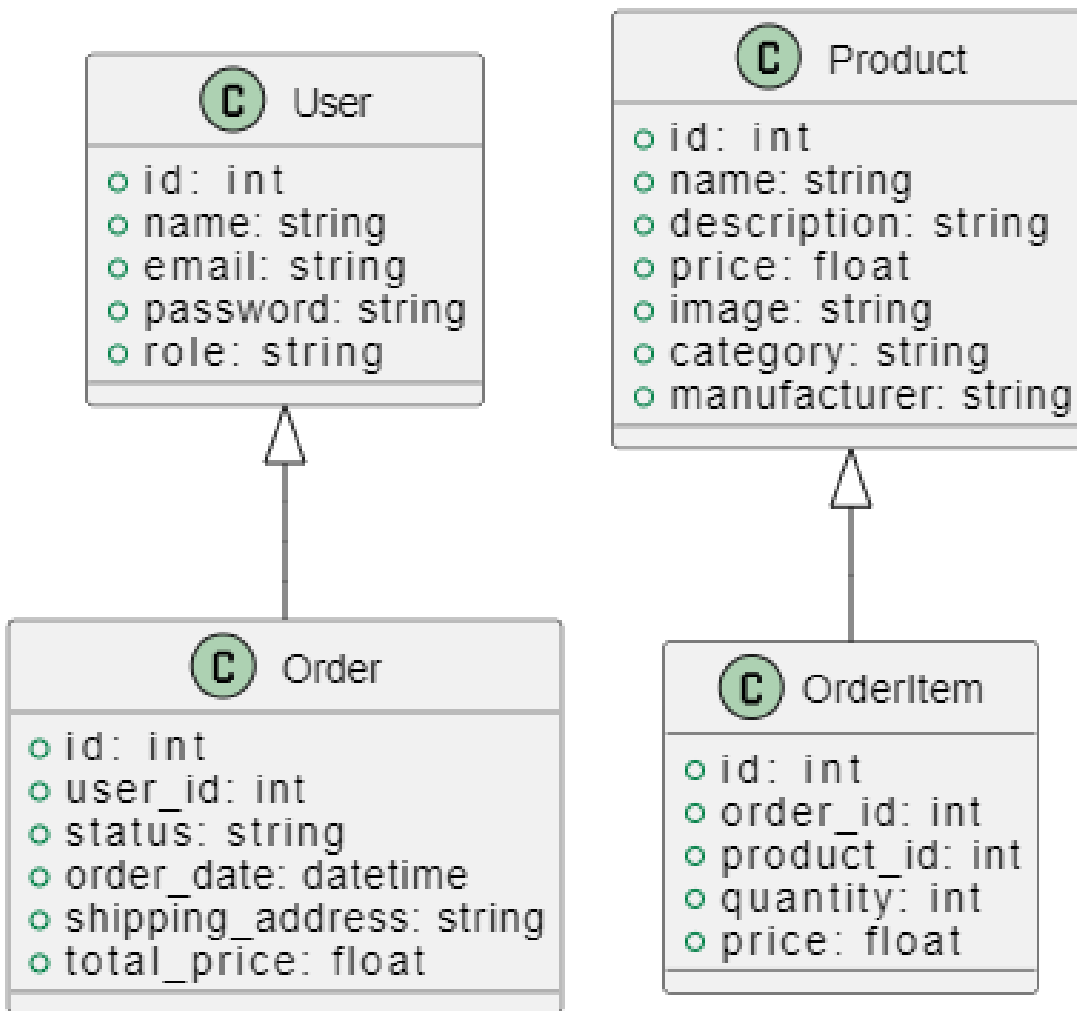
Secure Payment Gateway Integration:

- Integrate a trusted payment gateway to securely process online transactions.
- Implement industry-standard security measures to protect sensitive financial information, including data encryption and tokenization.

6. Use Case Diagram



7. UML Diagram



8. Architecture Design

Components

The website will be composed of the following main components:

- Frontend:
 - Built using React and Bootstrap 5
 - Responsible for handling user interactions, displaying product information, managing shopping carts, and processing checkouts
 - Communicates with the backend API through HTTP requests
- Backend:

- Built using Express and Node.js
- Responsible for handling data storage, product management, order processing, and user authentication
- Communicates with the MongoDB database for data persistence
- Database:
 - MongoDB will be used as the database due to its scalability and flexibility
 - Stores product data, user information, and order details

Deployment

The website will be deployed to a cloud platform such as AWS, Azure, or Google Cloud Platform. This will ensure high availability and scalability to handle the expected traffic.

Security

The website will implement industry-standard security measures to protect sensitive user data and financial information. These measures include:

- User authentication and authorization: Users will be required to create secure accounts and log in to access their information and place orders. Role-based access control (RBAC) will be implemented to restrict access to sensitive information based on user roles.
- Secure payment gateway integration: A trusted payment gateway will be integrated to securely process online transactions. Industry-standard security measures, such as data encryption and tokenization, will be implemented to protect sensitive financial information.
- Regular security audits and vulnerability scans: The website will be subjected to regular security audits and vulnerability scans to identify and address potential security weaknesses.

Performance

The website will be designed and optimized for performance to provide a fast and responsive user experience. This includes:

- Caching: Frequently accessed data will be cached to reduce server load and improve response times.
- Content delivery network (CDN): A CDN will be used to distribute static content, such as images and CSS files, to geographically dispersed servers, reducing latency and improving loading times for users worldwide.
- Load balancing: Load balancing will be implemented to distribute traffic across multiple servers, ensuring the website can handle peak traffic demands.

Monitoring

The website will be monitored to ensure its health and performance. This includes:

- Application monitoring: Metrics such as response times, error rates, and resource utilization will be monitored to identify and address potential issues promptly.
- Infrastructure monitoring: Infrastructure metrics such as CPU usage, memory consumption, and network traffic will be monitored to ensure the underlying infrastructure is performing optimally.

By following these guidelines, the Clarkton Corporation website will be a robust, scalable, and secure e-commerce platform that meets the needs of its target audience.

8.1 Business Logic Layer

The business logic layer (BLL) is responsible for implementing the core business rules and logic of the application. It acts as an intermediary between the presentation layer (frontend) and the data access layer (backend), ensuring that the application data is consistent, secure, and accessible.

Key Responsibilities of the BLL

- Data Validation: The BLL validates user input, product information, and order details to ensure data integrity and prevent invalid data from entering the system.
- Business Rule Enforcement: The BLL enforces business rules, such as product pricing, inventory management, and order processing guidelines, to maintain the consistency and integrity of the application's behavior.
- Data Transformation and Enrichment: The BLL may transform or enrich data before presenting it to the frontend or storing it in the database. This could involve tasks like formatting product descriptions, calculating taxes and shipping costs, or generating product recommendations.
- Business Process Orchestration: The BLL orchestrates complex business processes, such as order fulfillment, customer refunds, and product promotions, ensuring that these processes are executed correctly and in a timely manner.

Implementation Options

The BLL can be implemented using various techniques, including:

- Object-oriented programming (OOP): OOP provides a structured approach to encapsulating business logic within classes and objects, promoting modularity and maintainability.
- Service-oriented architecture (SOA): SOA promotes loose coupling between components and facilitates distributed computing, making it suitable for large-scale applications.
- Business rules engines (BREs): BREs provide a centralized repository for defining and managing business rules, enabling easy configuration and maintenance.

Benefits of a Strong BLL

A well-designed and implemented BLL offers several benefits, including:

- **Reduced Complexity:** The BLL abstracts away the complexity of business logic from the frontend and backend, making the application easier to develop, maintain, and understand.
- **Improved Data Integrity:** The BLL enforces data validation and business rules, ensuring the accuracy and consistency of application data.
- **Enhanced Maintainability:** The BLL encapsulates business logic in a centralized location, making it easier to modify and update business rules without affecting the overall structure of the application.
- **Increased Flexibility:** The BLL provides a layer of abstraction between the presentation and data access layers, allowing for easier integration with different technologies and frameworks.

8.2 Data Layer

The data layer is responsible for managing the persistence of data within the Clarkton Corporation website. It acts as an interface between the business logic layer (BLL) and the database, handling data storage, retrieval, and manipulation.

Key Responsibilities of the Data Layer

- **Data Storage:** The data layer stores product information, user data, order details, and other relevant data in a persistent manner, ensuring data integrity and availability.
- **Data Retrieval:** The data layer efficiently retrieves data based on requests from the BLL, providing the necessary information for business operations.
- **Data Manipulation:** The data layer performs CRUD (Create, Read, Update, Delete) operations on data, enabling the creation, modification, and deletion of data based on business logic requirements.
- **Database Connectivity:** The data layer manages the connection to the database, ensuring reliable and secure data access.

Data Storage Options

The data layer can utilize various data storage technologies, including:

- **Relational databases (RDBMS):** RDBMS provide a structured approach to data storage, making them suitable for storing and managing complex data relationships.
- **NoSQL databases:** NoSQL databases offer flexibility and scalability for storing unstructured or semi-structured data.

- Cloud-based databases: Cloud-based databases provide scalability, high availability, and reduced maintenance overhead.

Data Access Strategies

The data layer can employ various data access strategies, such as:

- Object-relational mapping (ORM): ORM frameworks like Hibernate or Entity Framework map database tables to object-oriented entities, simplifying data access and manipulation.
- Data access objects (DAOs): DAOs provide a layer of abstraction between the BLL and the database, encapsulating database-specific operations.
- Active record pattern: The active record pattern allows database objects to represent themselves in the application code, enabling direct manipulation of database objects.

Data Integrity and Security

The data layer must implement measures to ensure data integrity and security, including:

- Data validation: Data validation ensures that data entered into the system is accurate and consistent.
- Data type enforcement: Data type enforcement ensures that data is stored and manipulated according to its defined data type.
- Access control: Access control mechanisms restrict user access to sensitive data based on their roles and permissions.
- Data encryption: Data encryption protects sensitive data from unauthorized access.
- Data backup and recovery: Data backup and recovery ensure data availability in case of system failures or data corruption.

8.3 Deployment Layer

The deployment layer is responsible for deploying, managing, and monitoring the Clarkton Corporation website in a production environment. It ensures that the website is accessible to users, performs optimally, and scales to meet demand.

Key Responsibilities of the Deployment Layer

- Application Deployment: The deployment layer handles the deployment of the website's code and assets to the production environment. This involves tasks such as packaging the code, configuring deployment servers, and managing deployment workflows.

- **Infrastructure Management:** The deployment layer manages the underlying infrastructure, including servers, load balancers, and databases. This involves provisioning resources, configuring services, and ensuring high availability and scalability.
- **Monitoring and Alerting:** The deployment layer monitors the website's performance, resource utilization, and error rates. It generates alerts when potential issues arise and provides insights into the overall health of the application.
- **Version Control:** The deployment layer maintains a record of application changes and enables rollbacks to previous versions in case of issues.

Deployment Strategies

The deployment layer can employ various deployment strategies, including:

- **Blue-green deployments:** Blue-green deployments involve deploying a new version of the application alongside the existing version, switching traffic to the new version once testing is complete, and rolling back if necessary.
- **Rolling deployments:** Rolling deployments gradually deploy the new application version, replacing one server or a subset of servers at a time, minimizing downtime.
- **Canary deployments:** Canary deployments deploy the new application version to a small subset of users, monitor its performance and gather feedback before rolling out to the rest of the user base.

Deployment Tools

The deployment layer can utilize various tools to automate and manage the deployment process, such as:

- **Infrastructure as code (IaC) tools:** IaC tools like Terraform or Ansible automate the provisioning and configuration of infrastructure resources.
- **Continuous integration and continuous delivery (CI/CD) pipelines:** CI/CD pipelines automate the build, test, and deployment process, ensuring consistent and reliable deployment of new versions.
- **Configuration management tools:** Configuration management tools like Chef or Puppet manage the configuration of servers and applications, ensuring consistent configurations across the environment.

Security and Compliance

The deployment layer must implement security measures to protect the website and its data, including:

- **Network security:** Network security measures, such as firewalls and intrusion detection systems, protect the website from unauthorized access and cyberattacks.

- Data security: Data security measures, such as encryption and access control, protect sensitive data from unauthorized access or disclosure.
- Compliance with regulations: The deployment layer ensures compliance with relevant regulations, such as PCI DSS and GDPR, to protect user data and privacy.

9. Database Design

To effectively manage the data for the Clarkton Corporation website, a relational database schema is proposed. The schema consists of the following tables:

1. Users Table

SQL

```
CREATE TABLE users (
    user_id INT

PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT

NULL,
    email VARCHAR(255) NOT

NULL

UNIQUE,
    password VARCHAR(255) NOT

NULL,
    role VARCHAR(255) NOT

NULL

DEFAULT 'customer'
);
```

2. Products Table

SQL

```
CREATE TABLE products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    description TEXT NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    image_url VARCHAR(255) NOT NULL,
```

```
        category VARCHAR(255) NOT NULL,  
        manufacturer VARCHAR(255) NOT NULL,  
        stock_quantity INT NOT NULL  
    );
```

3. Orders Table

SQL

```
CREATE TABLE orders (  
    order_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    order_date DATETIME NOT NULL,  
    shipping_address TEXT NOT NULL,  
    order_status VARCHAR(255) NOT NULL,  
    total_price DECIMAL(10,2) NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES users(user_id)  
);
```

4. OrderItems Table

SQL

```
CREATE TABLE order_items (  
    order_item_id INT  
  
    PRIMARY KEY AUTO_INCREMENT,  
    order_id INT  
  
    NOT  
  
    NULL,  
    product_id INT  
  
    NOT  
  
    NULL,  
    quantity INT  
  
    NOT  
  
    NULL,  
    price DECIMAL(10,2)  
  
    NOT  
  
    NULL,  
    FOREIGN KEY (order_id) REFERENCES orders(order_id),  
    FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```

Data Relationships

The database schema defines the following relationships between the tables:

- One-to-Many Relationship between Users and Orders: A single user can place multiple orders.
- One-to-Many Relationship between Orders and OrderItems: A single order can contain multiple order items.
- Many-to-Many Relationship between Products and Orders: A product can be included in multiple orders.

Data Integrity Constraints

The database schema enforces data integrity constraints to ensure the accuracy and consistency of the data:

- Primary Key Constraints: Each table has a primary key attribute to uniquely identify each record.
- Foreign Key Constraints: Foreign key constraints ensure that references to other tables are valid.
- Unique Constraints: Unique constraints prevent duplicate values in specific columns, such as email addresses.

Normalization

The database schema is normalized to minimize data redundancy and improve data integrity. This ensures that data is stored in a structured and efficient manner, reducing the risk of data anomalies and inconsistencies.

10. Scalability

The Clarkton Corporation website project is designed to be scalable to accommodate the growing needs of the business. The following aspects of the project contribute to its scalability:

1. **MERN Stack:** The MERN stack (MongoDB, Express, React, and Node.js) is a well-known and widely used combination of technologies for building scalable and performant web applications. MongoDB's NoSQL structure provides horizontal scalability, allowing for seamless addition of data nodes to handle increasing data volumes. Express and Node.js enable efficient handling of concurrent requests and asynchronous operations, ensuring responsiveness under heavy traffic. React's component-based architecture promotes modularity and code reuse, facilitating maintainability and adaptability as the application grows.
2. **Cloud Deployment:** Deploying the website to a cloud platform like AWS, Azure, or Google Cloud Platform provides access to elastic infrastructure resources. Cloud platforms offer the ability to automatically scale up or down based on demand, provisioning additional resources when traffic spikes and scaling down when traffic subsides. This dynamic allocation of resources optimizes resource utilization and costs while maintaining performance.
3. **Caching:** Implementing caching strategies can significantly improve the website's scalability by reducing the load on the database and backend servers. Frequently accessed data, such as product information and user profiles, can be cached in memory or on a distributed cache system, reducing the need to repeatedly query the database. This caching mechanism can significantly reduce response times and improve overall performance.
4. **Content Delivery Network (CDN):** Utilizing a CDN can further enhance the website's scalability and global reach. A CDN distributes static content, such as images, CSS files, and JavaScript files, across geographically dispersed servers. This distribution ensures that users worldwide can access static content from the nearest CDN node, reducing latency and improving loading times.
5. **Load Balancing:** Implementing load balancing techniques can distribute incoming traffic across multiple servers, preventing any single server from becoming overloaded. Load balancers can distribute requests based on various factors, such as server load, request type, and user location, ensuring optimal resource utilization and maintaining a responsive user experience under heavy traffic.
6. **Monitoring and Optimization:** Continuously monitoring the website's performance and resource utilization is crucial for identifying and addressing potential scalability bottlenecks. Tools like Prometheus, Grafana, and New Relic can provide valuable insights into application metrics, such as response times, error rates, and CPU usage. By analyzing these metrics, developers can proactively identify and address performance issues before they impact user experience.

By employing these strategies and adopting a cloud-based approach, the Clarkton Corporation website can effectively scale to accommodate the growing needs of the business, ensuring a responsive and performant e-commerce platform for its users.

11. Security

Ensuring the security of the Clarkton Corporation website is paramount to protecting user data, financial information, and the overall integrity of the e-commerce platform. Implementing robust security measures across all layers of the application is crucial to safeguard against cyberattacks and maintain user trust.

User Authentication and Authorization

1. **Secure User Registration and Login:** Implement strong password policies, including minimum length, complexity requirements, and password hashing to protect user credentials.
2. **Role-Based Access Control (RBAC):** Enforce RBAC to restrict user access to sensitive data and functionalities based on their roles and permissions.
3. **Multi-Factor Authentication (MFA):** Consider implementing MFA to add an extra layer of security for critical actions, such as financial transactions or account management.

Data Protection and Privacy

1. **Data Encryption:** Encrypt sensitive data, such as user passwords, payment card information, and personal details, both at rest and in transit.
2. **Data Access Controls:** Implement granular access controls to restrict database access to authorized users and applications.
3. **Data Masking and Pseudonymization:** Consider masking or anonymizing sensitive data for testing or development purposes to protect privacy.

Application Security

1. **Input Validation and Sanitization:** Validate user input to prevent malicious code injection (SQL injection, cross-site scripting) and ensure data integrity.
2. **Regular Security Audits and Vulnerability Scans:** Conduct regular security audits and vulnerability scans to identify and address potential weaknesses in the application code and infrastructure.
3. **Secure Coding Practices:** Follow secure coding practices, such as using secure libraries, avoiding common coding vulnerabilities, and keeping software components up to date.

Infrastructure Security

1. **Secure Network Configuration:** Implement secure network configurations, including firewalls, intrusion detection systems, and network segmentation, to protect against unauthorized access and cyberattacks.
2. **Secure Server Configuration:** Harden server configurations by disabling unnecessary services, patching vulnerabilities promptly, and applying strong access controls.
3. **Regular Security Updates:** Maintain up-to-date operating systems, software components, and third-party libraries to minimize vulnerabilities.

Incident Response and Recovery

1. **Incident Response Plan:** Develop a comprehensive incident response plan to effectively identify, contain, and remediate security incidents in a timely manner.
2. **Regular Security Awareness Training:** Provide regular security awareness training to employees to educate them about common cybersecurity threats and best practices.
3. **Data Backup and Recovery:** Maintain regular backups of critical data to enable recovery in case of system failures or data breaches.

12. API Documentation

All API endpoints require authentication using JSON Web Tokens (JWTs). To obtain a JWT, users must first register and log in to the website. Once authenticated, the JWT can be included in the Authorization header of API requests.

API Endpoints

1. Products

- GET /products: Retrieves a list of all products.
- GET /products/:id: Retrieves a specific product by its ID.
- POST /products: Creates a new product.
- PUT /products/:id: Updates an existing product.
- DELETE /products/:id: Deletes an existing product.

2. Orders

- GET /orders: Retrieves a list of all orders for the current user.
- GET /orders/:id: Retrieves a specific order by its ID.
- POST /orders: Creates a new order.
- PUT /orders/:id: Updates an existing order.
- DELETE /orders/:id: Deletes an existing order.

3. Users

- GET /users/:id: Retrieves a specific user by their ID.
- PUT /users/:id: Updates an existing user.

Error Handling

All API endpoints return a JSON response with a status code indicating the outcome of the request. In case of errors, the response will include an error message and error code.

Example Usage

To retrieve a list of all products, send a GET request to the /products endpoint:

```
curl -X GET https://clarktoncorporation.com/api/products
```

To retrieve a specific product by its ID, send a GET request to the /products/:id endpoint, replacing :id with the actual product ID:

```
curl -X GET https://clarktoncorporation.com/api/products/1234
```

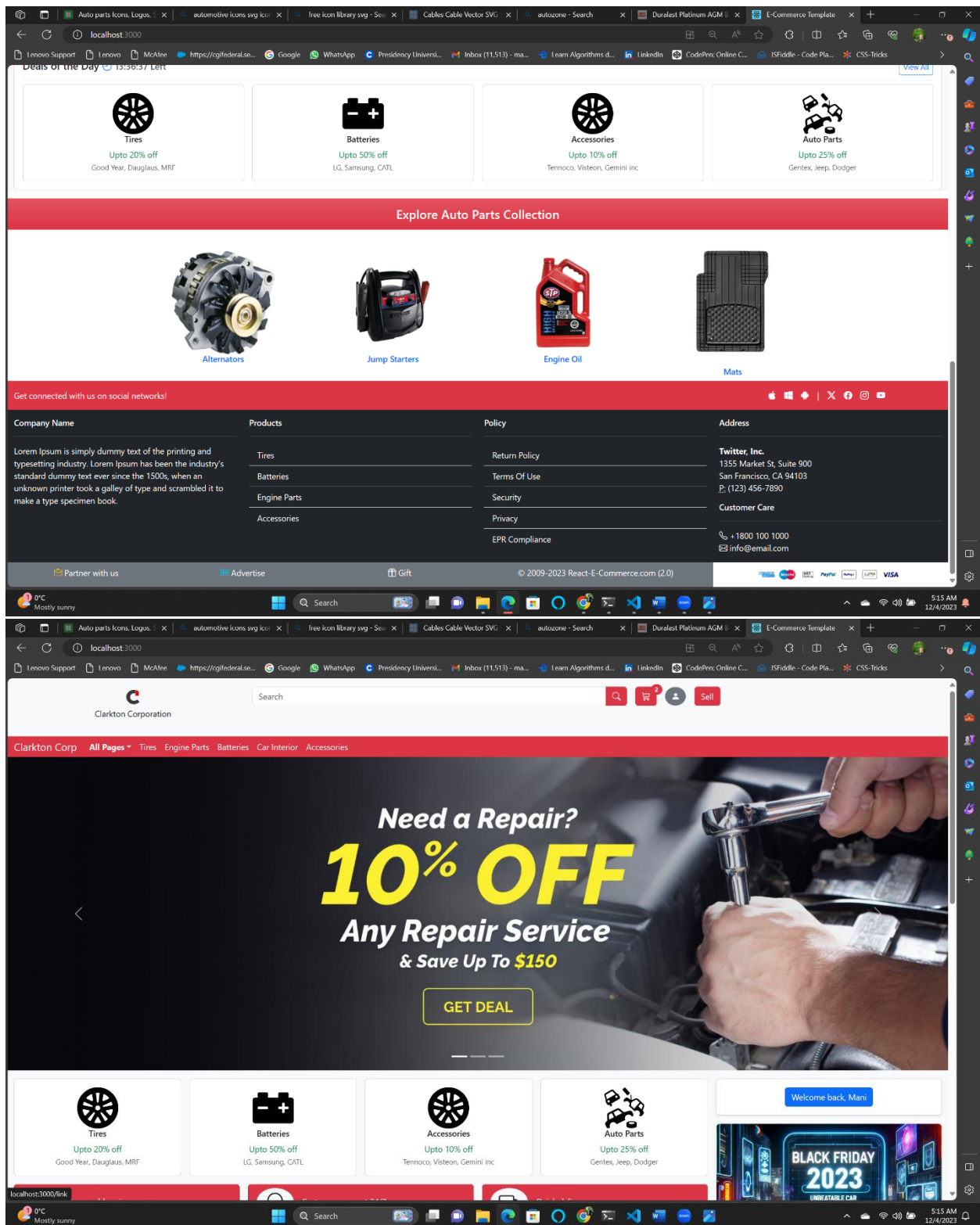
To create a new product, send a POST request to the /products endpoint, including the product data in the request body:

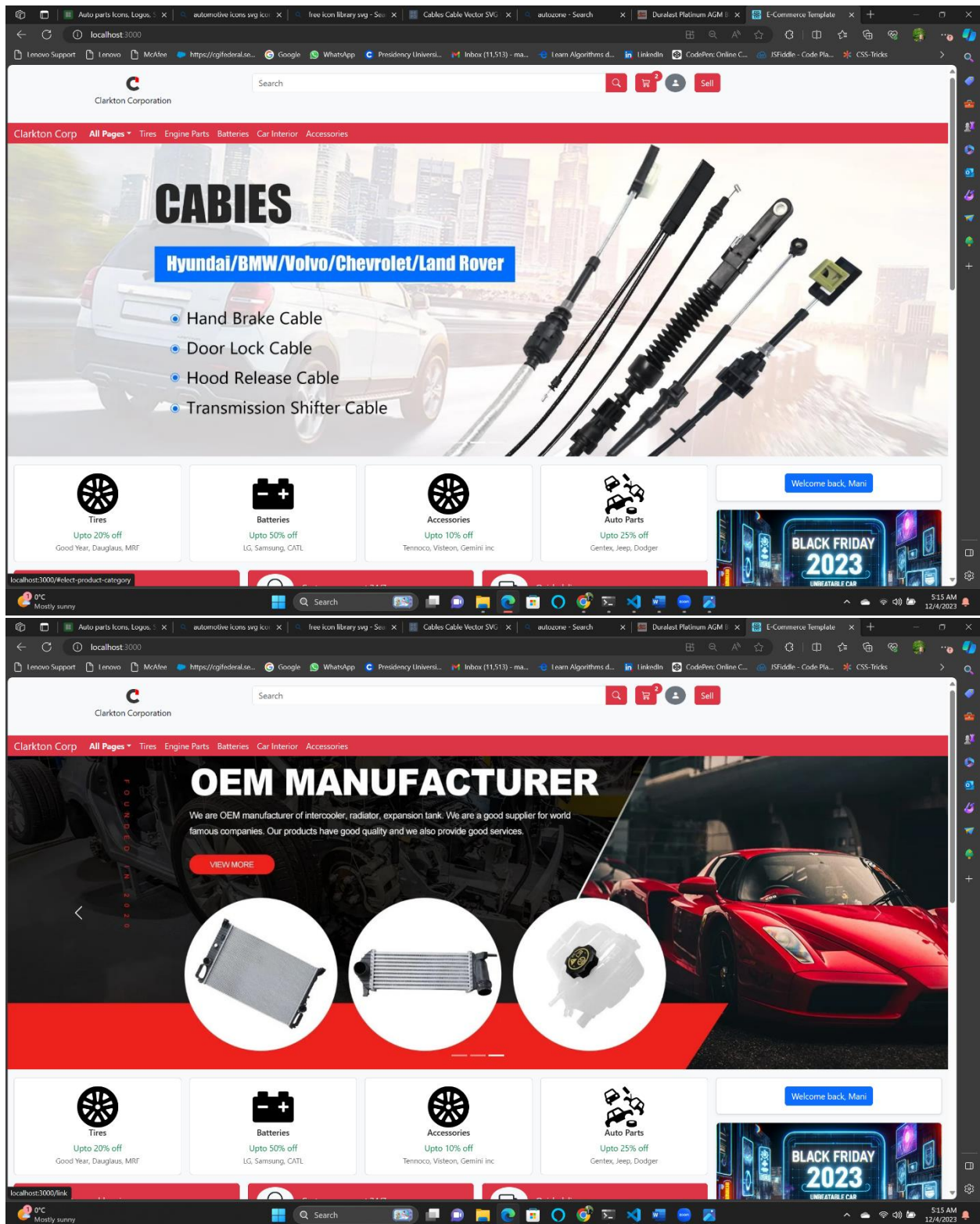
```
curl -X POST https://clarktoncorporation.com/api/products \
-H 'Authorization: Bearer <your-jwt>' \
-H 'Content-Type: application/json' \
-d '{
  "name": "Product Name",
  "description": "Product Description",
```

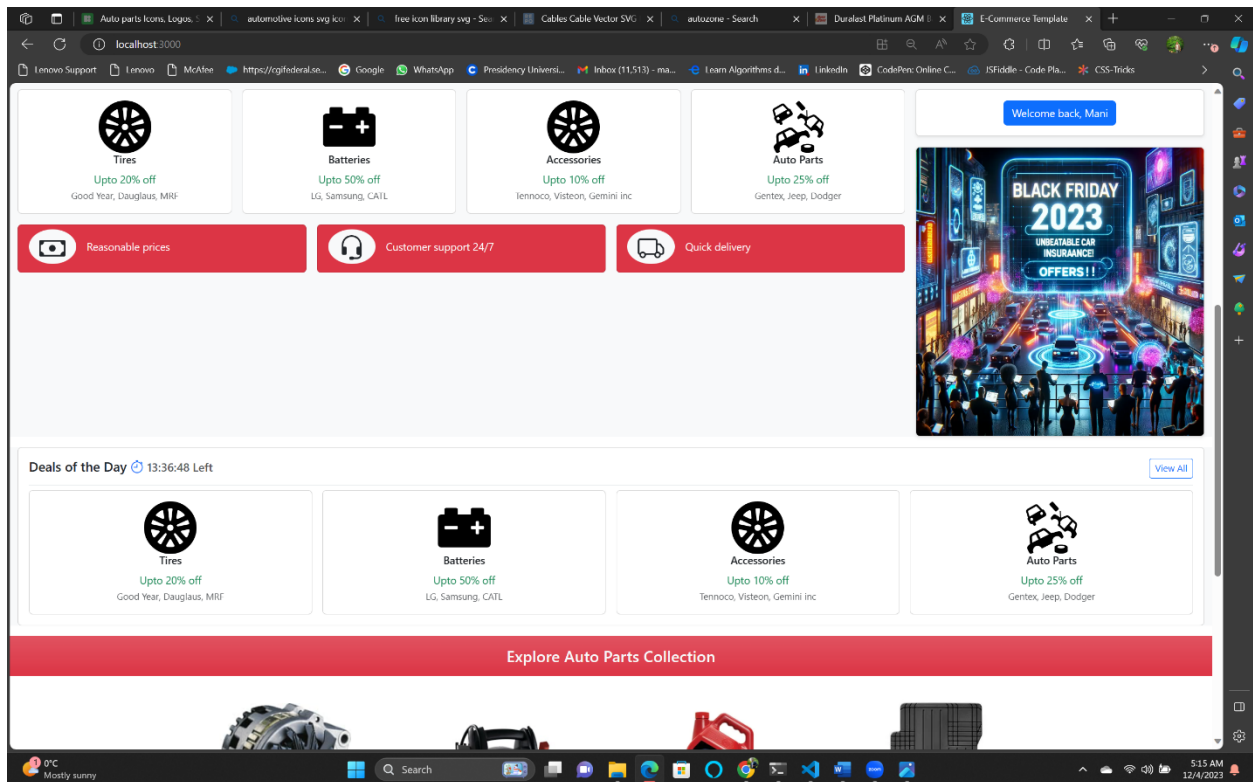
```
"price": 12.99,  
"image_url": "https://image.com/product.jpg",  
"category": "Category",  
"manufacturer": "Manufacturer"  
}
```

13. User Interfaces

Homepage







Products List Page

Clarkton Corporation

Search

W

2

Sell

Clarkton CorpAll PagesTiresEngine PartsBatteriesCar InteriorAccessories

Tires

Home / Tires / Tires

128 results for "tires"

Most Popular

88

CATEGORIES

Tires

Engine Parts

Batteries

Windshield

Car Interiors

Accessories

PRICE

\$24.00 - \$29.00 (4)

\$33.00 - \$35.00 (2)

\$70.00 - \$99.00 (5)

CUSTOMER REVIEW

★★★★★

★★★★☆

★★★☆☆

★★☆☆☆

★☆☆☆☆

Good Year Tires 215/195/R160 95V

New

★★★★★

Nulla sodales sit amet orci eu vehicula. Curabitur metus velit, fermentum a velit ac, sodales egestas lacus. Etiam congue velit vel luctus dictum. Pellentesque at pellentesque sapien.

\$180

\$200

-10%

Free shipping

W

General 215/195/R160 95V Winter Tires

Hot

★★★★★

Maecenas suscipit volutpat gravida. Nulla hendrerit nisi a lectus blandit aliquam. Integer enim magna, consequat sed justo nec, auctor sagittis urna.

\$475

Free shipping

W

Michelin 215/195/R160 95V Winter Tires

★★★★★

- Morbi lobortis velit non consectetur porta.
- Duis auctor risus ac purus vehicula tempor.
- Fusce at ipsum a leo tempor malesuada.
- Curabitur tincidunt justo vel volutpat suscipit.

\$10

W

Good Year Tires 215/195/R160 95V

New

★★★★★

Nulla sodales sit amet orci eu vehicula. Curabitur metus velit, fermentum a velit ac, sodales egestas lacus. Etiam congue velit vel luctus dictum. Pellentesque at pellentesque sapien.

\$180

\$200

-10%

Free shipping

W

1

2

3

4

5

6

7

Next

15

0°C

Mostly sunny

Search

5:17 AM

12/4/2023

Product Detail Page

Clarkton Corporation

Search

🔍 🛒 👤 Sell

Clarkton Corp

All Pages


Tires

Engine Parts

Batteries

Car Interior

Accessories



Good Year Tires 215/195/R160 95V

★★★★★ | 42 ratings and 4 reviews

Availability: In stock

Sold by: Authorised Store

\$1900

\$2000

\$100

1

+

+

+

Add to cart

Buy now

👤

Product Highlights

- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Etiam ullamcorper nibh eget faucibus dictum.
- Cras consequat felis ut vulputate porttitor.

Details

Ratings & Reviews

Questions and Answers

Shipping & Returns

Size Chart

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

▶ Even more details

▶ Even more details

Description lists

A description list is perfect for defining terms.

Eulmod

Vestibulum id ligula porta felis euismod semper eget lacina odio sem. Donec id elit non mi porta gravida at eget metus.

Malesuada porta

Ktiam porta sem malesuada magna mollis euismod.

FEATURED PRODUCTS



Good Year Tires 215/195/R160 95V

★★★★★

\$180

\$200



General 215/195/R160 95V Winter Tires

★★★★

\$475



Wintercraft 215/195/R160 95V All Season

★★★

\$1900

\$2000



Michelin 215/195/R160 95V Winter Tires

★★★★

\$10

Details

Ratings & Reviews

Questions and Answers

Shipping & Returns

Size Chart

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

▶ Even more details

▶ Even more details

Description lists

A description list is perfect for defining terms.

Eulmod

Vestibulum id ligula porta felis euismod semper eget lacina odio sem. Donec id elit non mi porta gravida at eget metus.

Malesuada porta

Ktiam porta sem malesuada magna mollis euismod.

CUSTOM SERVICE



Free Delivery

From \$39.89



Support 24/7

Online 24 hours



Free return

965 a day

Get connected with us on social networks!

🍏 🍎 🍓 🍷 🍷 🍷 🍷 🍷 🍷

Company Name

Products

Policy

Address

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Tires

Batteries

Engine Parts

Accessories

Return Policy

Terms Of Use

Security

Privacy

EPR Compliance

Twitter, Inc.

1355 Market St, Suite 900

San Francisco, CA 94103

P: (123) 456-7890

Customer Care

📞 +1800 100 1000

✉ info@email.com

Partner with us

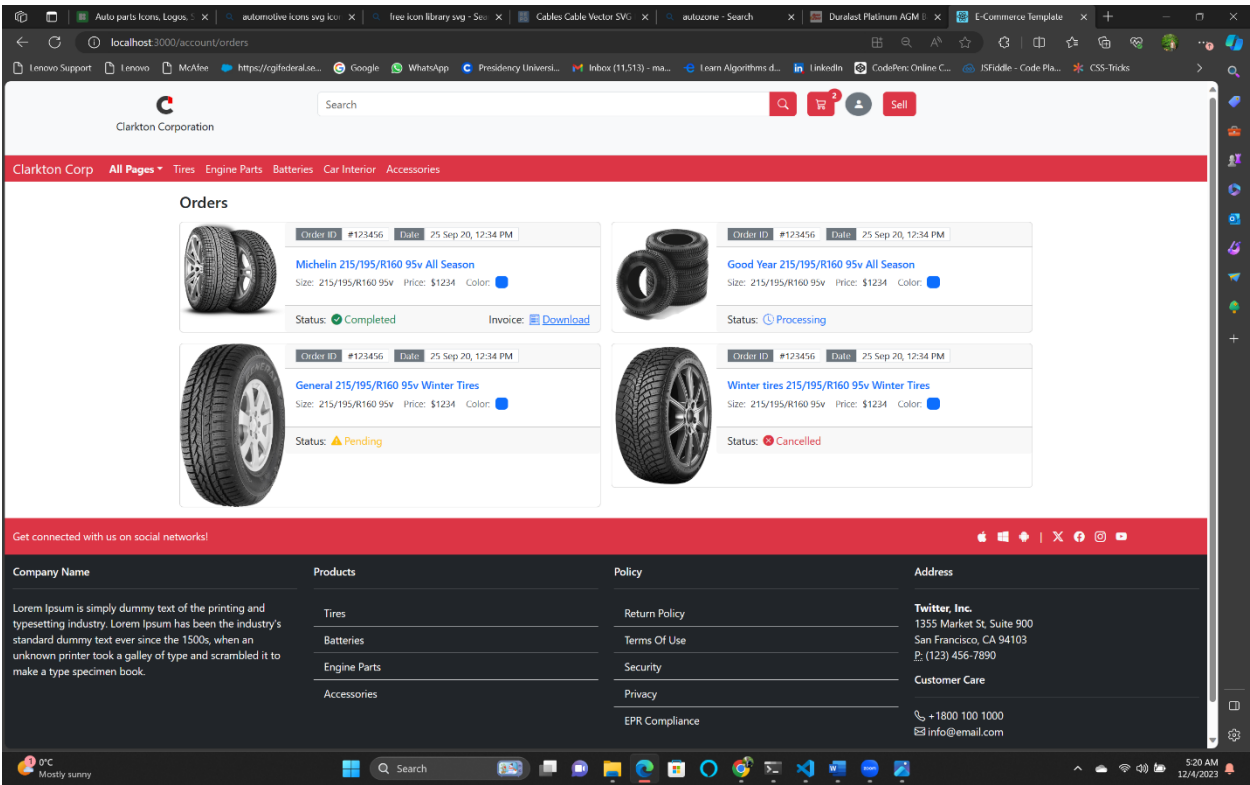
Advertise

Gift

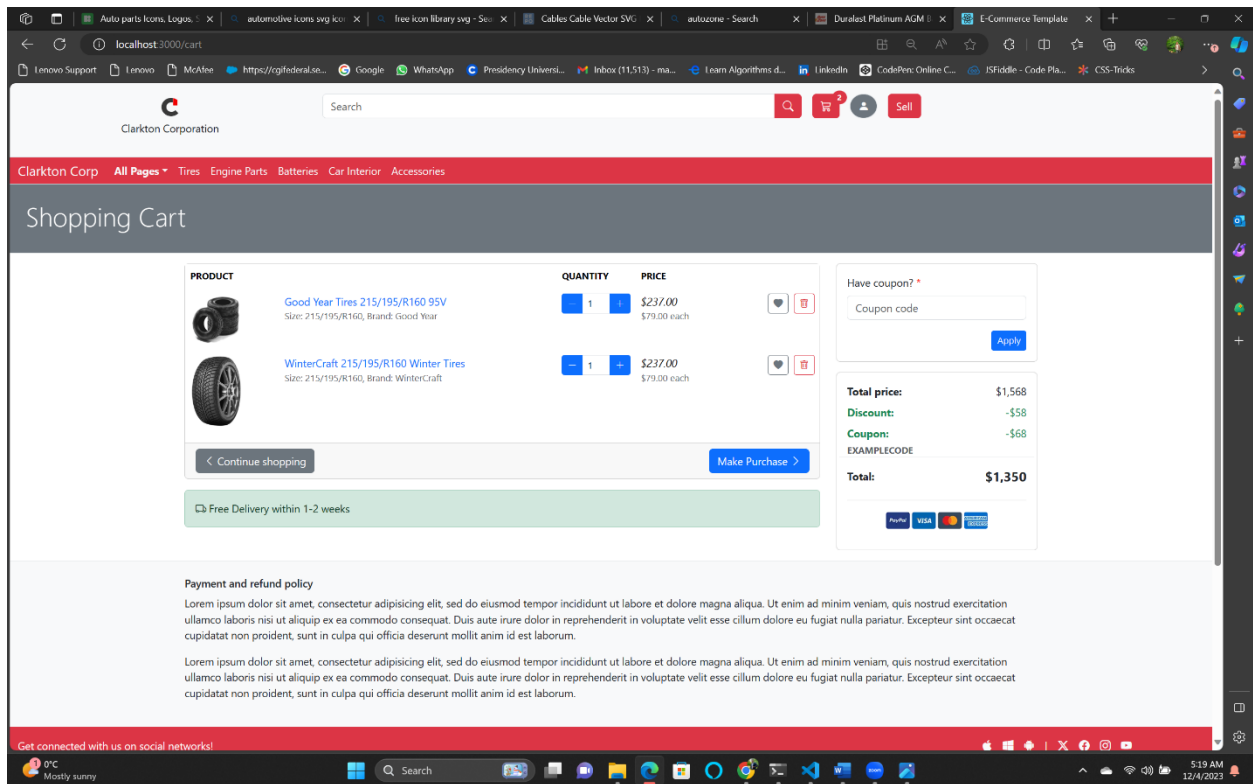
© 2009-2023 React-E-Commerce.com (2.0)

🍏 🍎 🍓 🍷 🍷 🍷 🍷 🍷 🍷

Orders Page



Cart Page



Sign Up Form Page

The image shows a web browser window with multiple tabs open. The active tab is 'localhost:3000/account/signup'. The page displays a 'Sign Up' form with the following fields:

- First Name
- Last Name
- Username
- Password
- Contact Number
- Address
- Profile Image (with a 'Choose File' button and a file named 'car-tire-png-457.png' selected)
- Email ID
- Date of Birth (with a placeholder 'mm/dd/yyyy' and a calendar icon)

Below the form is a blue 'Submit' button. At the bottom of the page is a red footer bar with the text 'Get connected with us on social networks!' and icons for various social media platforms. Below the footer bar is a navigation menu with links for 'Company Name', 'Products', 'Policy', and 'Address'. The Windows taskbar at the bottom shows the date and time as 6:29 AM on 12/4/2023.

Sign In Page

The screenshot displays a web browser window with multiple tabs open. The active tab shows a sign-in page for Clarkton Corporation. The page layout includes a header with the company logo and name, a search bar, and a navigation menu. The main content area features a sign-in form with fields for 'Mobile no' and 'Your password', a 'Log In' button, and links for 'Create your account' and 'Forgot password?'. Below the form, there are social media icons for X, Facebook, and Google+. The footer contains a grid of links for 'Company Name', 'Products', 'Policy', and 'Address', along with a 'Partner with us' section and a copyright notice.

Clarkton Corporation

Search

Clarkton Corp All Pages Tires Engine Parts Batteries Car Interior Accessories

Sign In

Mobile no *

Mobile no without country code

Your password *

Log In

[Create your account](#) [Forgot password?](#)

Or you can join with

X Facebook Google

Get connected with us on social networks!

Company Name Products Policy Address

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Tires Batteries Engine Parts Accessories

Return Policy Terms Of Use Security Privacy EPR Compliance

Twitter, Inc.
1355 Market St, Suite 900
San Francisco, CA 94103
P: (123) 456-7890
Customer Care
+1800 100 1000
info@email.com

Partner with us Advertise Gift © 2009-2023 React-E-Commerce.com (2.0)

0°C Mostly sunny 5:23 AM 12/4/2023

User Profile Page


Clarkton Corporation

Search

🔍 🛒 👤 ⚙️

Clarkton Corp All Pages Tires Engine Parts Batteries Car Interior Accessories

Profile Detail



No image.

Choose File No file chosen

You don't allow uploading a photo more than 5MB

With supporting text below as a natural lead-in to additional content.

📎 Your name

📎 Mobile no without country code

Change Password

Current Password *

New Password *

Confirm New password *

Submit

Setting

🔔 Notification

🔇

📧 E-Mail and SMS

🔇

% Promotional offers E-Mail and SMS

🔇

Delete Account

Submit

Saved Card

+

🗑️ 123 *** 123

🗑️

🗑️ 123 *** 123


🗑️

🗑️ 123 *** 123

🗑️

0°C Mostly sunny

Search



5:21 AM

12/4/2023

Sell Product Form Page

The screenshot displays a web browser window with multiple tabs open. The active tab shows a form titled "Sell Auto Parts" on the Clarkton Corporation website. The form has the following fields:

- Product Name:** A text input field.
- Category:** A text input field.
- Price:** A text input field.
- Description:** A large text area for product details.
- Product Image:** A file upload section with a "Choose File" button and a "No file chosen" status.

Below the form is a "Submit" button. The footer of the page includes a red banner with social media icons and a dark grey section with the following content:

Company Name	Products	Policy	Address
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.	Tires	Return Policy	Twitter, Inc. 1355 Market St, Suite 900 San Francisco, CA 94103 P: (123) 456-7890 Customer Care
	Batteries	Terms Of Use	
	Engine Parts	Security	

The browser's address bar shows "localhost:3000/sellform". The Windows taskbar at the bottom indicates the date and time as 5:22 AM on 12/4/2023.

14. Developer Documentation

Documentation provides comprehensive information for developers who want to integrate with the Clarkton Corporation website or contribute to its development.

Technology Stack

The Clarkton Corporation website is built using the following technologies:

- Frontend: React, Bootstrap 5
- Backend: Express, Node.js, MongoDB
- Deployment: AWS Elastic Container Registry (ECR), AWS Elastic Kubernetes Service (EKS)
- Continuous Integration/Continuous Delivery (CI/CD): Jenkins

Development Environment Setup

To set up the development environment for the Clarkton Corporation website, follow these steps:

1. Install Node.js: Download and install the latest version of Node.js from <https://nodejs.org/en/download/current>.
2. Install MongoDB: Download and install MongoDB from <https://www.mongodb.com/try/download/community>.
3. Clone the project repository: Clone the Clarkton Corporation website repository from GitHub.
4. Install dependencies: Install the project's dependencies using npm or yarn:

npm install

5. Start the development server: Start the development server using the following command:

npm start

API Integration

The Clarkton Corporation website exposes a RESTful API that allows developers to integrate with the website and its functionalities. The API documentation can be found at

<https://github.com/CoolKit-Technologies/eWeLink-API>.

Contributing to the Project

The Clarkton Corporation website is an open-source project, and we welcome contributions from developers. To contribute to the project, please follow these steps:

1. Fork the project repository: Fork the Clarkton Corporation website repository on GitHub.
2. Create a feature branch: Create a new branch for your development work.
3. Make changes: Make the desired changes to the project's code.
4. Commit your changes: Commit your changes with meaningful commit messages.

5. Create a pull request: Create a pull request from your feature branch to the main branch of the forked repository.
6. Address feedback: Address any feedback or comments from the project maintainers.
7. Merge your pull request: Once your pull request is approved, merge it into the main branch of the forked repository.

Additional Resources

- Project GitHub Repository: <https://github.com/mani978/clarktonwebsite/tree/master>
- API Documentation:
<https://github.com/mani978/clarktonwebsite/blob/main/Response-RFP.docx>
- Write mail to us: <mailto:manikanta9780@gmail.com>