**eCommerce Website Documentation**

**Introduction**

This documentation provides an overview of an eCommerce website built using the MERN stack, which consists of the following technologies:

* MongoDB
* Express.js
* React.js
* Node.js.

This stack enables the development of a robust and scalable eCommerce platform with a rich user interface.

**Backend supportive libraries**

**Backend development of the eCommerce website utilizes several supportive libraries. Here are some of the key libraries used:**

* **bcrypt** (version 5.1.0): Used for hashing and encrypting user passwords for secure storage.
* **cookie-parser** (version 1.4.6): Parses and handles cookies in the Express.js application.
* **cors** (version 2.8.5): Enables Cross-Origin Resource Sharing to allow requests from different domains.
* **dotenv** (version 16.0.3): Loads environment variables from a .env file into the application.
* **express** (version 4.18.2): Web application framework for building the backend API.
* **jsonwebtoken** (version 9.0.0): Generates and verifies JSON Web Tokens (JWT) for user authentication and authorization.
* **mongoose** (version 7.0.3): Object Data Modeling (ODM) library for MongoDB to simplify database operations.
* **multer** (version 1.4.5-lts.1): Handles file uploads in the backend, such as product images.
* **nodemailer** (version 6.9.1): Library for sending emails, used for features like order confirmation and password reset.
* **nodemon** (version 2.0.22): Development utility that automatically restarts the server upon file changes, improving the development workflow.
* Additionally, the backend integrates with the Stripe payment gateway for handling secure online payments.

**Front-End supportive libraries**

**frontend development of the eCommerce website, several supportive libraries are used. Here are some of the key libraries used:**

* **tailwindcss** :Utility-first CSS framework for creating responsive and custom designs.
* **mui/material** :Material-UI library for building responsive and aesthetic user interfaces.
* **axios** : Promise-based HTTP client for making API requests from the frontend.
* **reduxjs/toolkit**: Toolkit for simplifying Redux state management in the application.
* **stripe/react-stripe-js**: React components for integrating Stripe payments in the frontend.
* **stripe/stripe-js**: Stripe JavaScript library for client-side payment processing.
* **country-state-city** : Library for retrieving country, state, and city data for address forms.
* **framer-motion** : Animation library for creating smooth and interactive UI transitions.
* **react-icons** : Library for easily including popular icon packs in the application.
* **react-lottie** :Integration of Lottie animations in React applications.
* **react-router-dom**: Library for declarative routing and navigation in React applications.
* **react-toastify** :Toast notification library for displaying informative messages to users.
* **socket.io-client** : Client-side library for real-time bidirectional communication between the server and the frontend.
* **timeago.js**: Library for formatting timestamps as relative time (e.g., "5 minutes ago").

**Backend functionalities**

**The backend of the eCommerce website encompasses various functionalities to support the overall system. Here are the key functionalities implemented in the backend:**

**User Authentication and Authorization:** Provides user registration, login, and authentication using JSON Web Tokens (JWT). Implements authorization checks to ensure that only authenticated users can access protected routes.

**Shop Creation and Management:** Allows shop owners or administrators to create and manage their own shops within the eCommerce platform. This includes features like creating a shop, updating shop details, managing shop inventory, and setting up shop policies.

**Middleware**: Implements middleware functions to handle common tasks such as request logging, error handling, input validation, and authentication verification. Middleware functions are used to enhance the functionality and security of the API endpoints.

**Coupon Code Creation:** Implements functionality to create and manage coupon codes. This feature allows shop owners to offer discounts to customers by generating coupon codes with various configurations, such as percentage-based or fixed amount discounts.

**Discount Events:** Manages discount events that are applicable to specific products or shop-wide. Discount events can include seasonal sales, flash sales, or promotional offers. It tracks the availability and application of discounts to eligible products during the event period.

**Product Management:** Enables the management of products available on the eCommerce platform. Includes features like creating, updating, and deleting products, as well as fetching product details and listing products.

**Order Management**: Handles the processing of customer orders. Includes features such as placing orders, retrieving order history, updating order status, and generating invoices.

**Shopping Cart**: Manages the user's shopping cart functionality. Allows adding products to the cart, updating quantities, removing items, and calculating the total price.

**Payment Processing**:

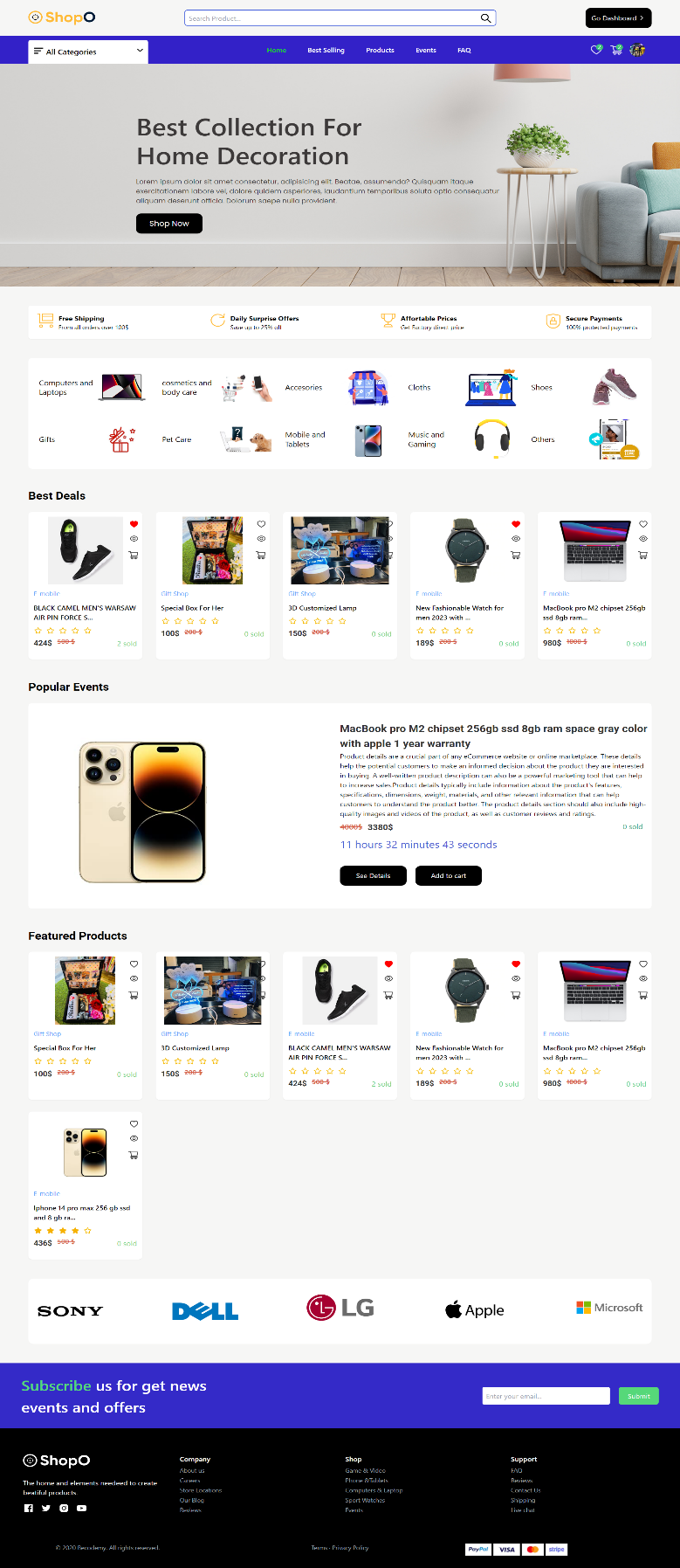
Integrates with a payment gateway (such as Stripe) to securely process online payments for customer orders. Handles payment confirmation and updates order status accordingly.

Cash on Delivery: Provides an alternative payment method for customers who prefer to pay cash upon delivery of their orders. It handles the order status and notifies the shop owner or delivery personnel to collect cash payment during order fulfillment.

**File Upload:** Supports the uploading and management of product images or other media files using libraries like Multer. Enables efficient storage and retrieval of file resources.

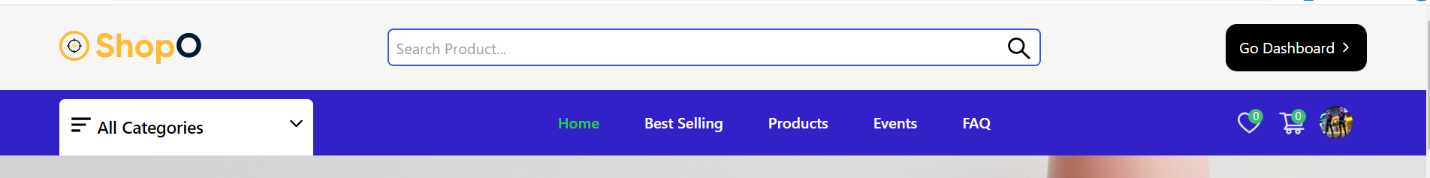
**Real-Time Communication**: Implements real-time features such as live chat using Socket.IO. Allows users to interact with each other or receive instant updates about their orders.

**Front-End functionalities**

****

**Main Home Page**

**Header and navigation sections:**



**Search Bar**: Displays a search bar in the header section, allowing users to enter keywords and search for products based on their query. The search results are displayed in a dropdown, providing relevant product suggestions.

**Create Shop Option:** Provides a "Become Seller" button or link in the header, allowing users to create their own shops within the eCommerce platform. This feature is typically available to shop owners or authorized users.

**All Categories of Product**: Includes a dropdown menu that lists all available categories of products. Users can select a category to browse products within that specific category.

**Navbar**: The navbar typically includes links to home, All products, Best Selling, Events, FAQ pages and user profile.

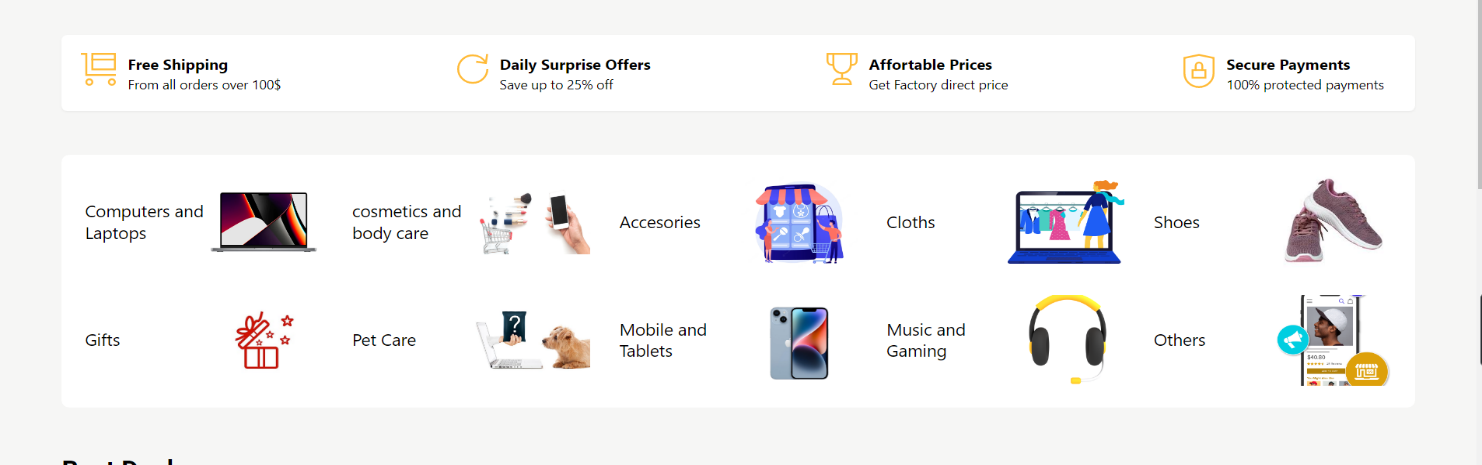
**Add to Cart Icon**: Users can click this button to add the selected product to their shopping cart.

**Wishlist Icon:** Users can add products to their wishlist, creating a collection of desired items for future reference or purchase.

**Profile Icon**: Displays a profile icon or user avatar in the header, representing the user's account. Clicking on the icon opens a dropdown menu with options like user profile, order history, settings, and logout.

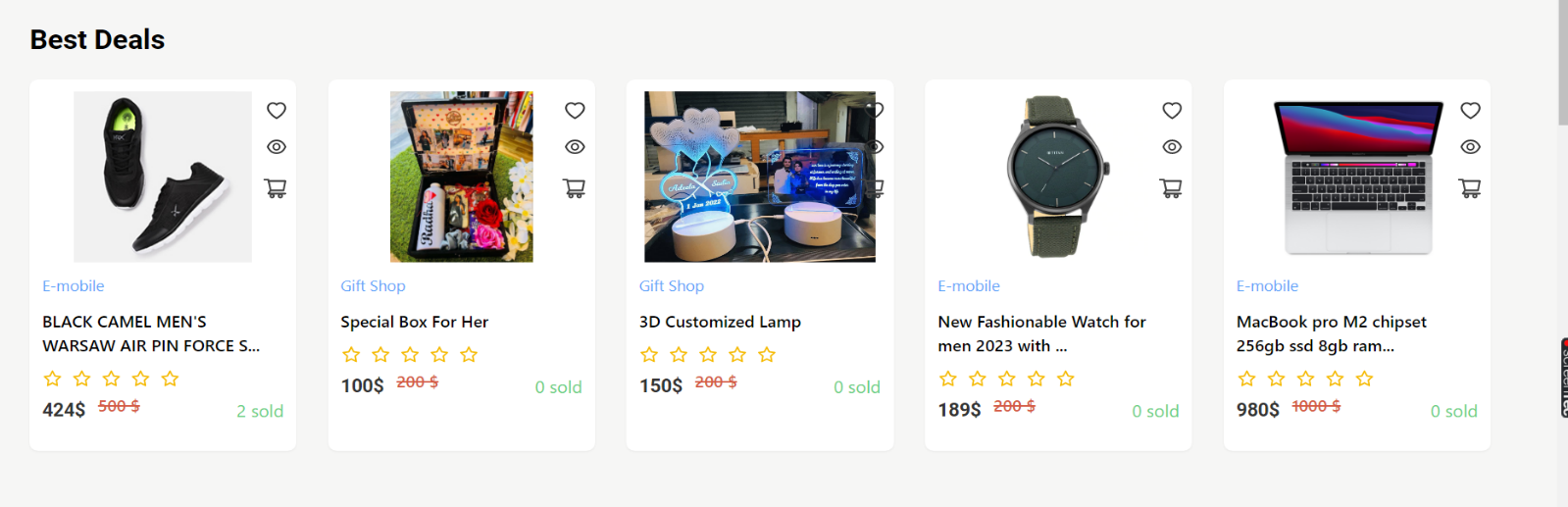
**Category Section**:

Separate Section for each category where users can browse products specific to that category. When a category is clicked from the homepage, it should navigate to the corresponding category page.

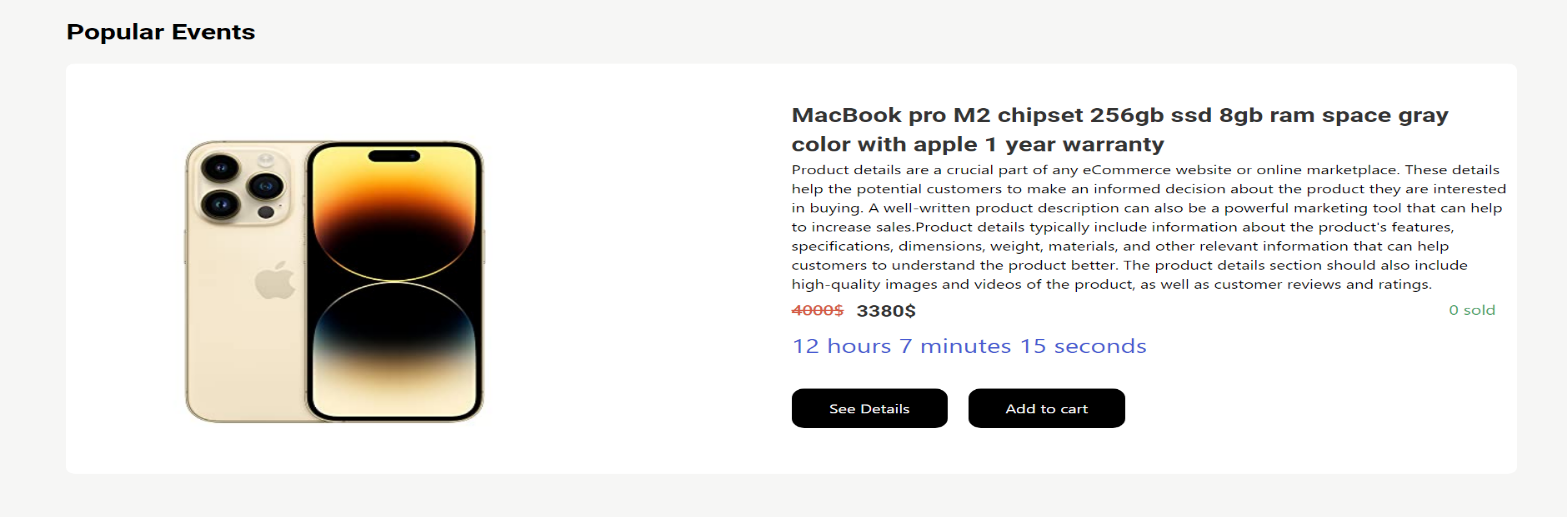


**Best Selling Section**

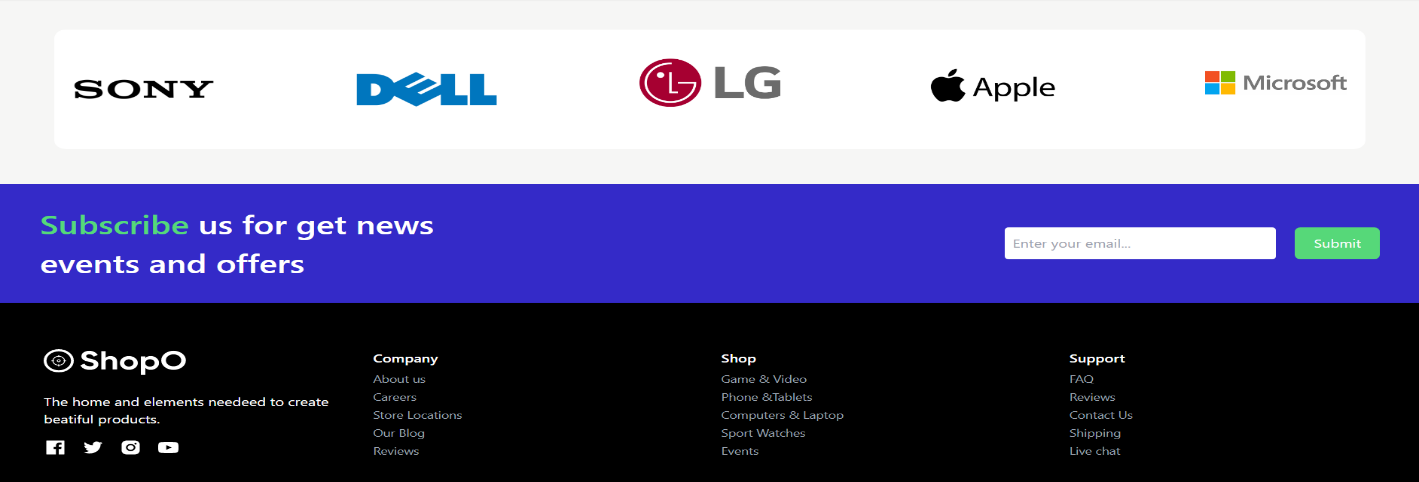
Best selling section that showcases products based on their popularity or sales,



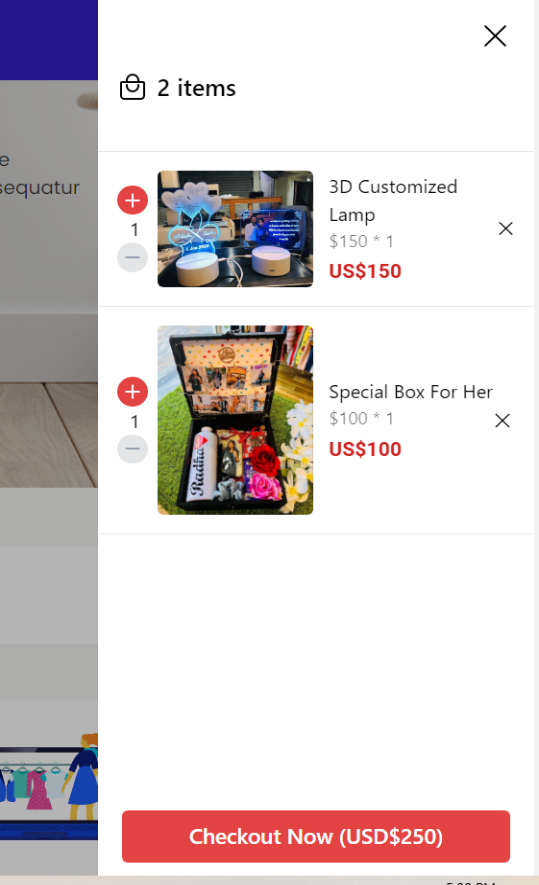
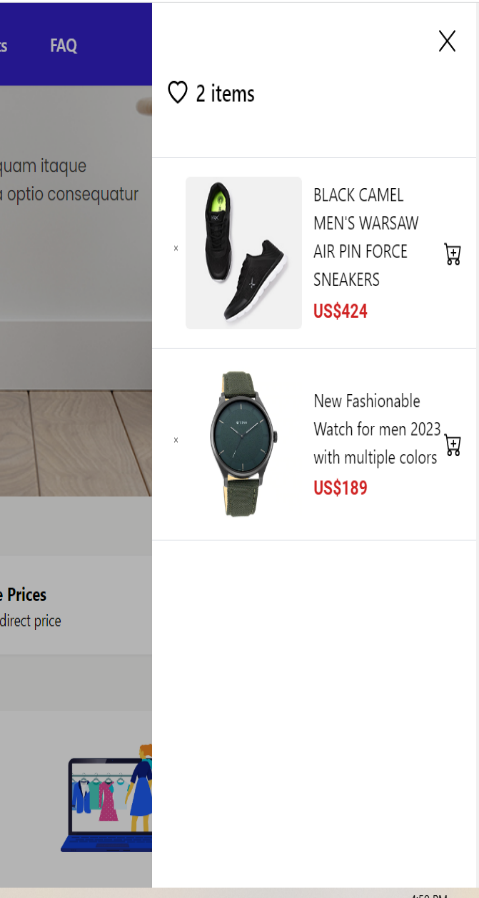
**Popular Events**

****

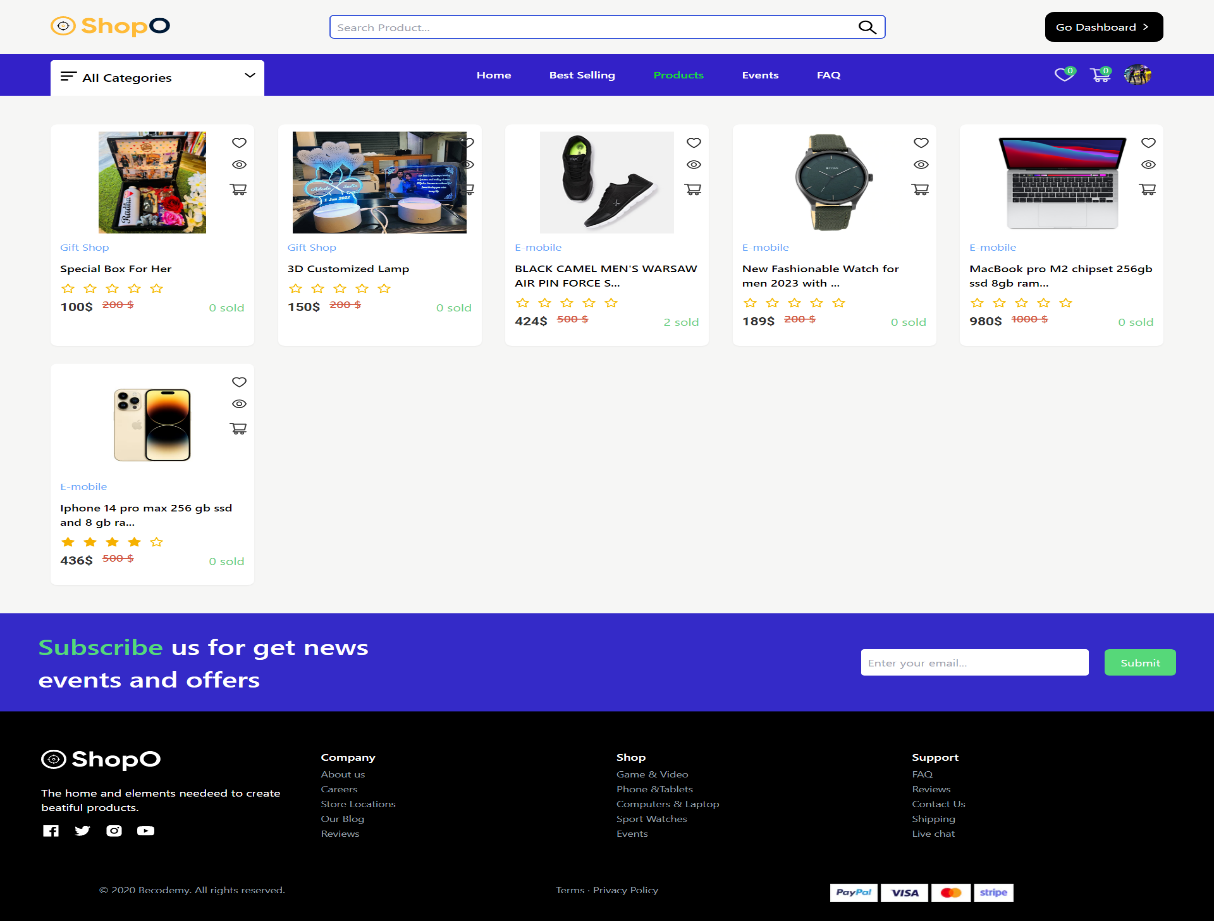
**Footer**

****

**WishList Section Add To Cart Section**

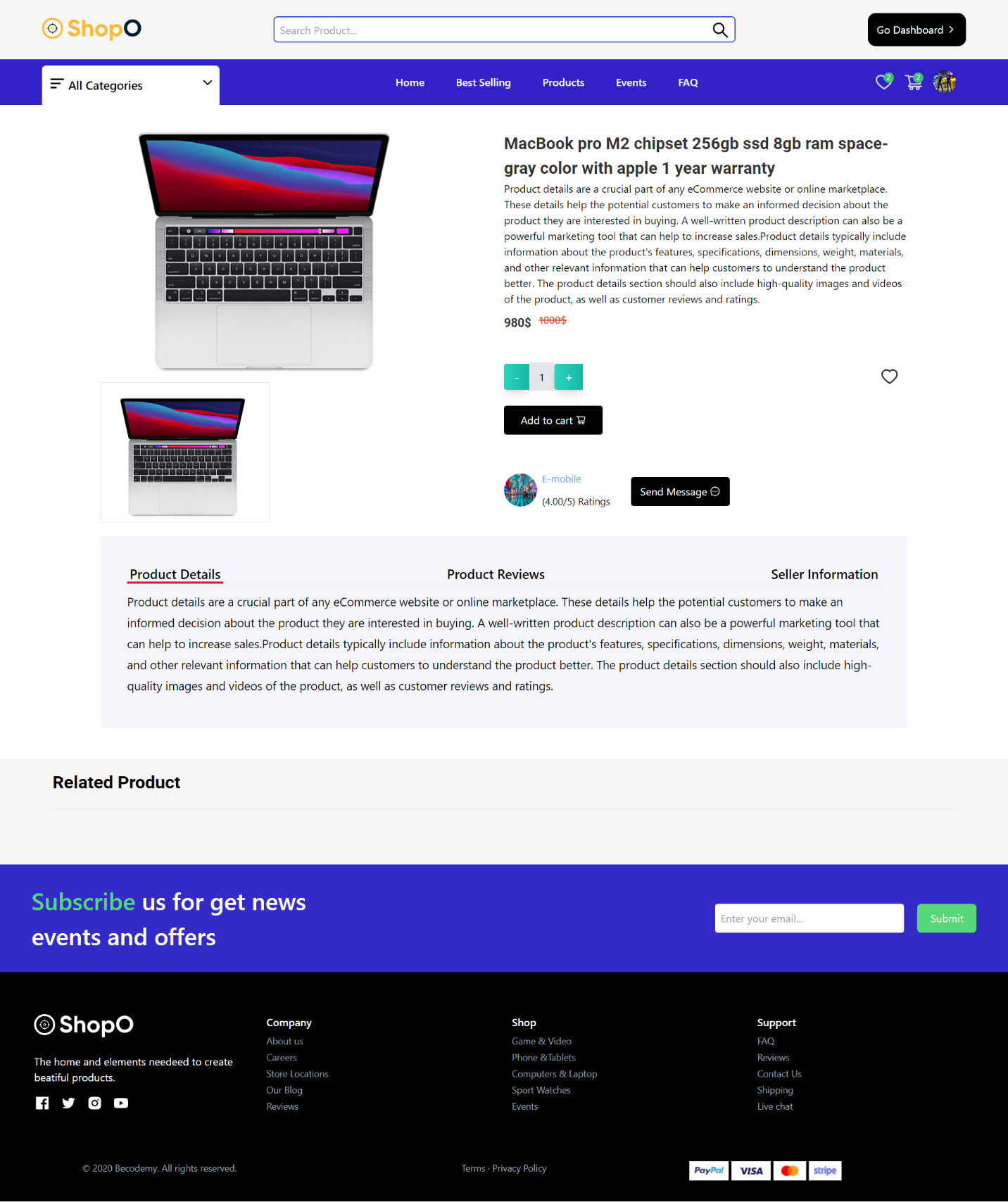
****

**All Products Page**

****

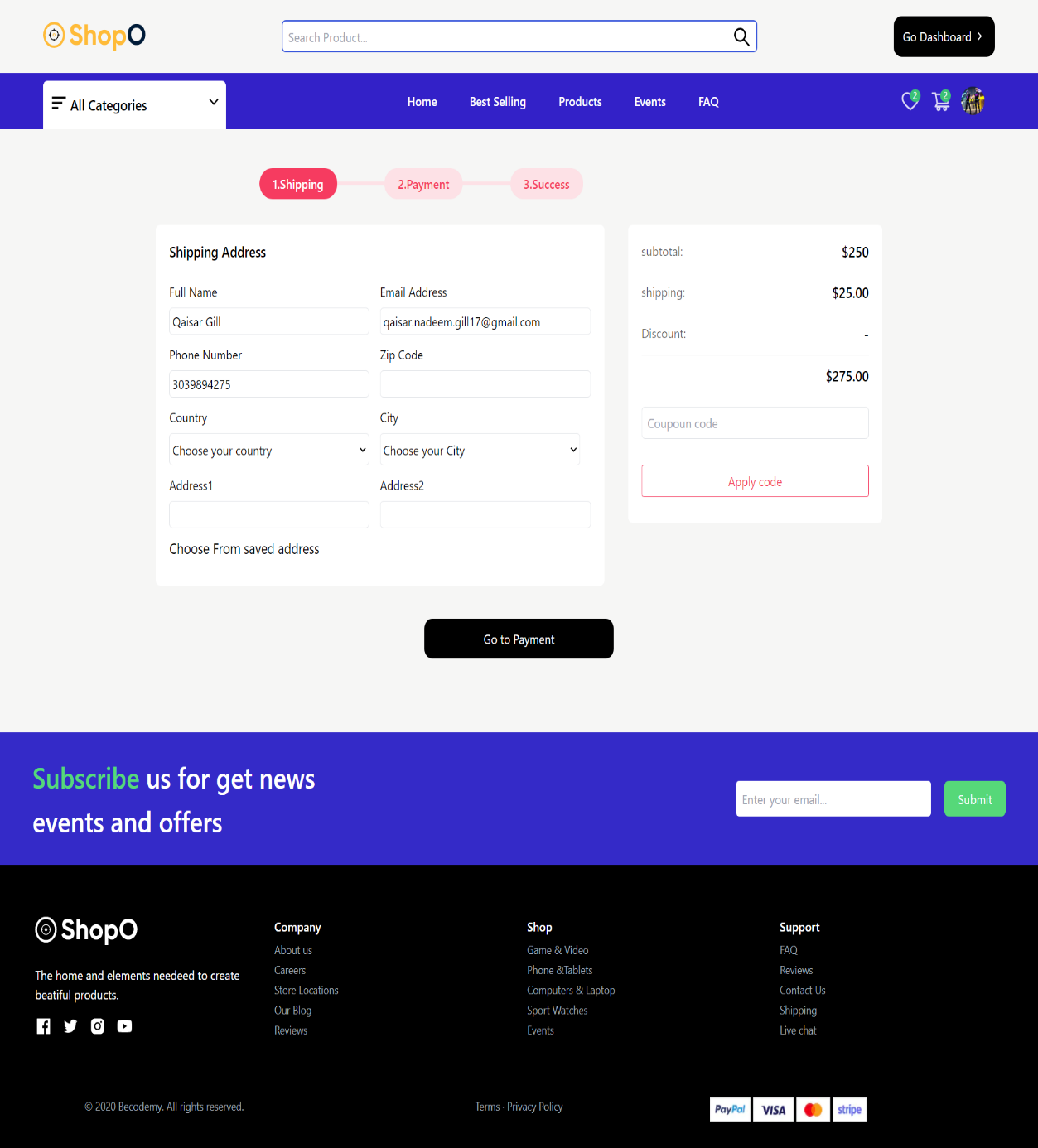
**Product Details:**

Displays detailed information about a selected product, including images, description, specifications, and customer reviews. Allows users to add the product to the shopping cart from the product details page.



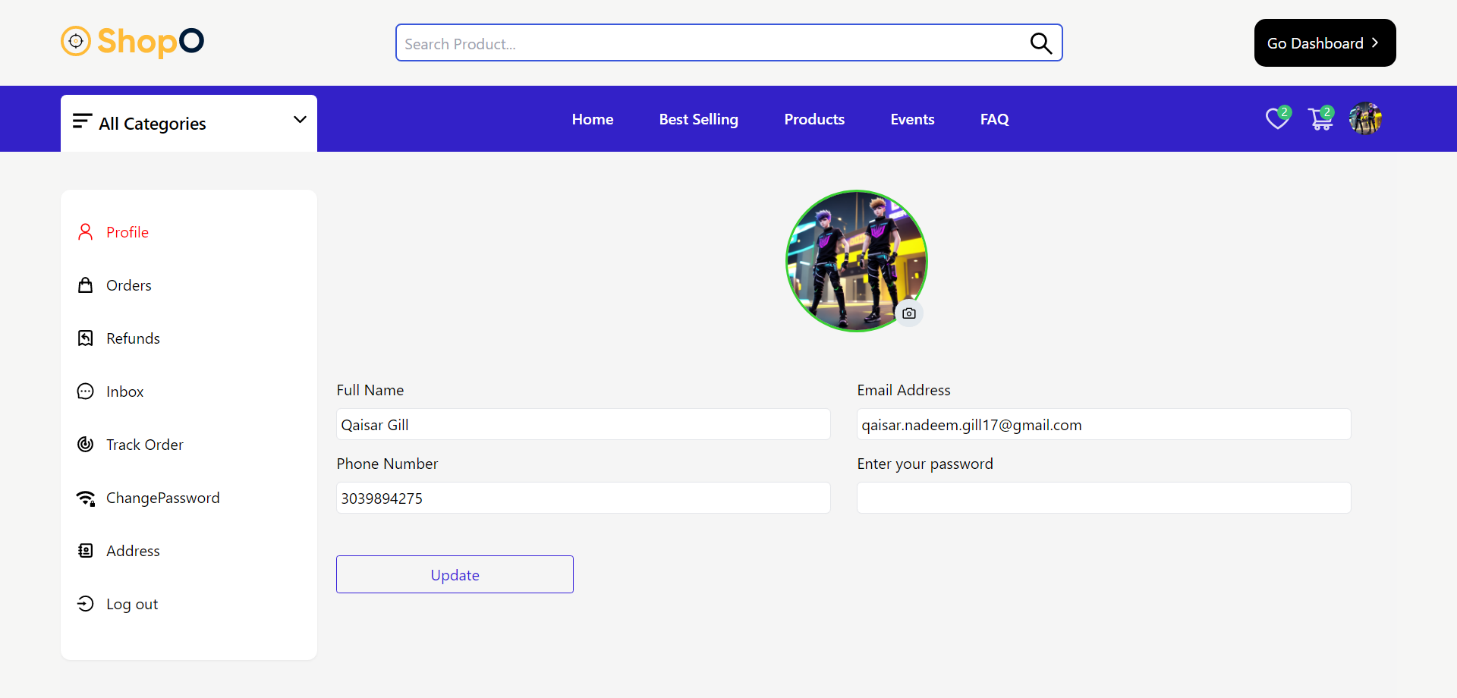
**Checkout Page**

Checkout and Payment: Guides users through the checkout process, collecting shipping and billing information. Integrates with a payment gateway (e.g., Stripe , cash on delivery) to securely process online payments. Provides feedback on the payment status and generates order confirmation.

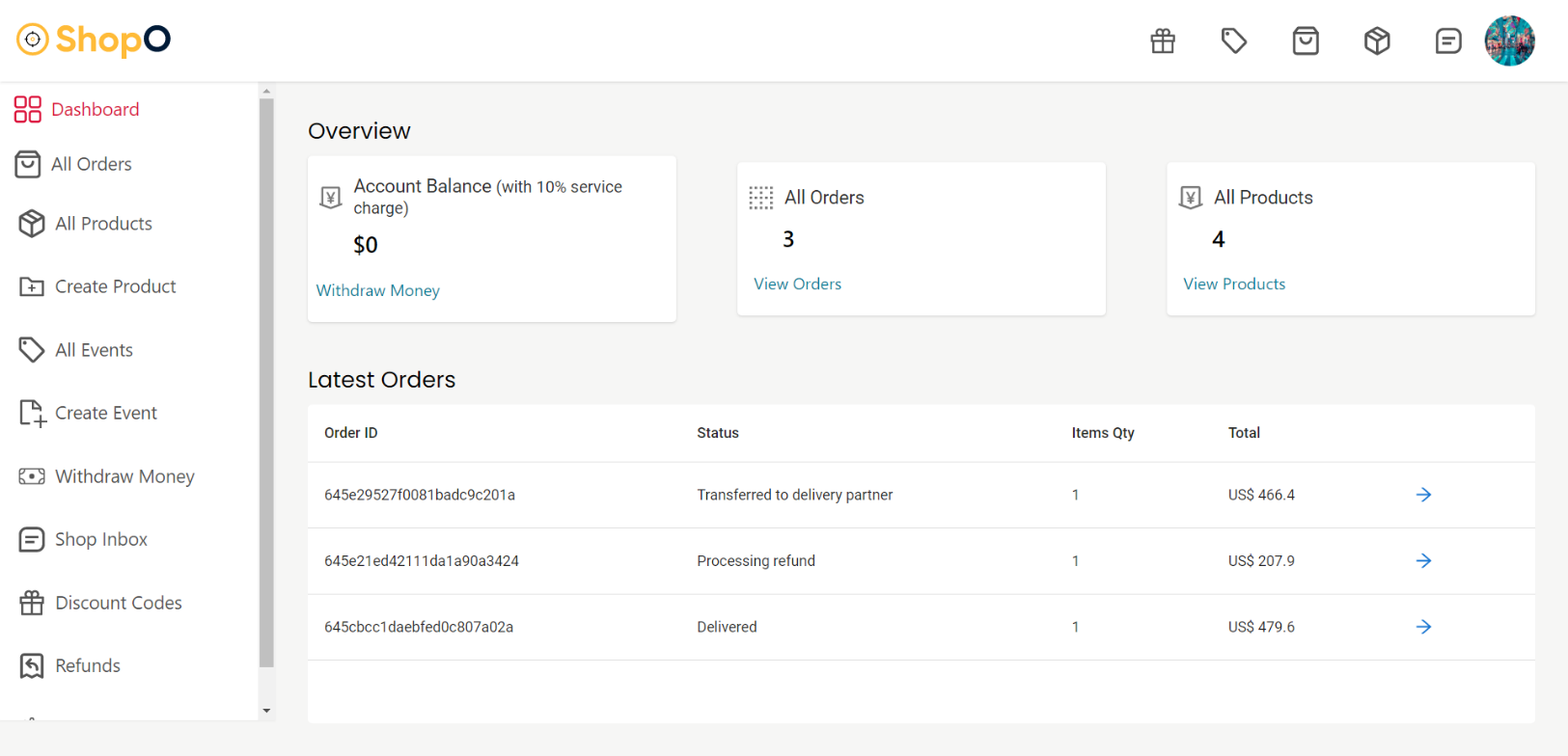
****

**User Profile Dashboard**:

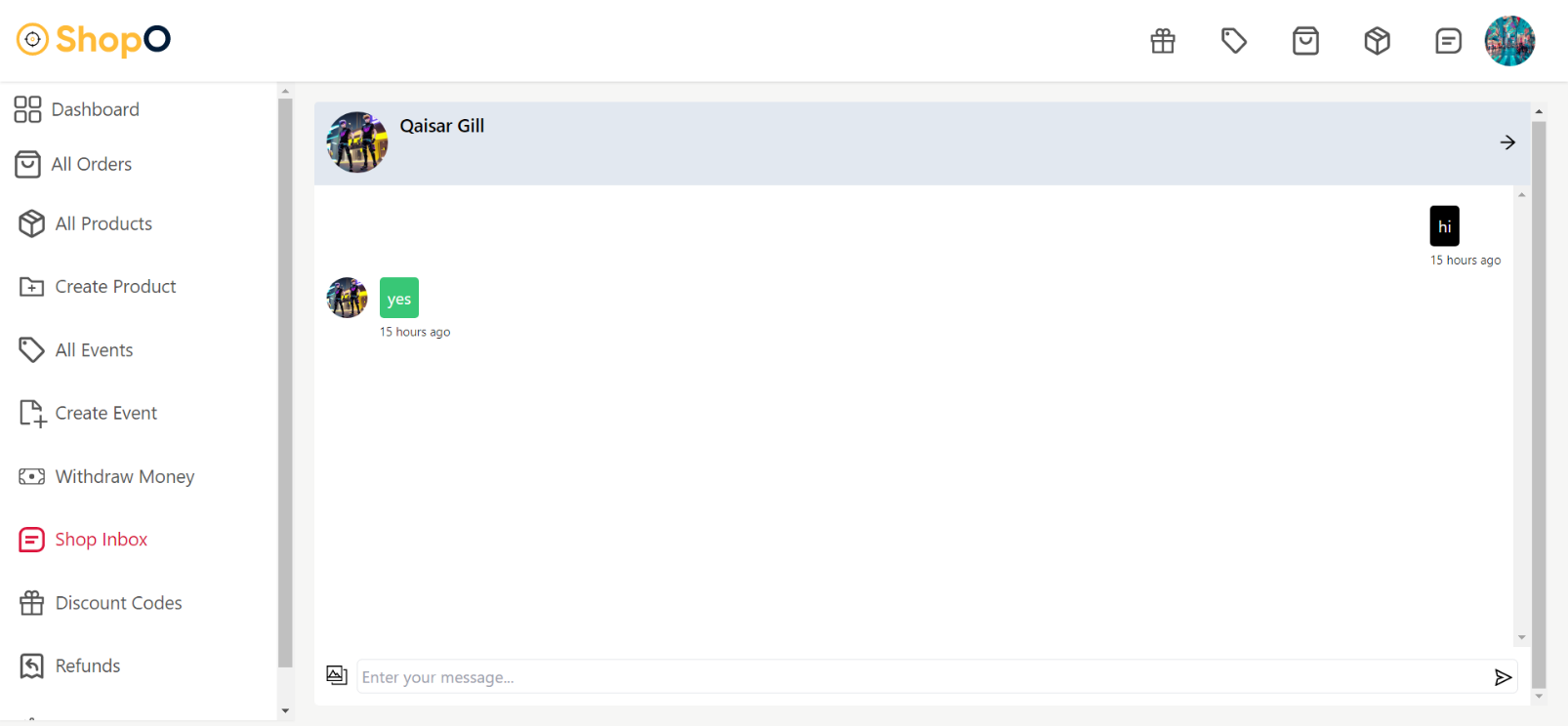
Provides a user profile page where users can view and update their personal information, such as name, email address, and shipping address. Allows users to manage their saved payment methods and update their account settings.

****

**Seller or Shop owner Dashboard**  
The Seller or Shop Owner Dashboard provides a dedicated interface for sellers to manage their shops, products, orders, and other aspects related to their online business. Here are some key features and functionalities typically found in a Seller or Shop Owner Dashboard:

****

**Chat and Inbox Page**

****