Title:LocalMarketHub - A MultiVendor Ecommerce Platform

Project Description:

LocalMarketHub is a multi-vendor ecommerce website designed to connect users with nearby markets, providing a seamless shopping experience tailored to the user's geographical location. The platform aims to bridge the gap between online shopping convenience and the charm of local markets, allowing users to discover and purchase products from vendors within their vicinity.

Key Features:

Location-Based Product Discovery:

- Users can easily search for products based on their current location, enabling them to explore nearby markets and discover unique offerings from local vendors.

Vendor Integration:

- Local businesses and vendors can join the platform, creating their virtual storefronts to showcase their products. Vendors can manage inventory, process orders, and engage with customers through the platform.

Dynamic Search and Filters:

- The platform incorporates dynamic search algorithms that consider both product relevance and proximity to the user. Advanced filters allow users to refine their search based on categories, pricing, and vendor ratings.

Distance Sorting:

- Search results are sorted based on the distance between the user and the vendor, ensuring that users see the most relevant and nearby options first. This enhances the local shopping experience and reduces delivery times.

Front-End:

User Access:

Home Page:

- Explore a dynamic homepage showcasing a variety of products from different vendors.
- Conveniently access "Login," "Signup," "Cart," and "Wishlist" buttons.

Product Search by Market:

- Easily search for products by selecting the preferred market name, streamlining the shopping experience.Using Google Maps to allocate the user to nearest shop

Product Details:

- Click on a product to view detailed information, including the vendor's shop name.
- Explore the vendor's shop to discover additional products.

User Interaction:

- Comment on purchased products, providing valuable feedback.
- Add desired products to the "Wishlist" and "Cartlist" for future reference and easy checkout.

Product Categories:

- Explore specific product categories like "Best Selling," "Men's," "Women's," "Electronics," and "Events."

Q&A Section:

- Engage in a Q&A section for inquiries and additional product information.

Shop Owner Access:

Vendor Dashboard:

- Access a dedicated dashboard for the shop, managing orders and inventory efficiently.

Order Tracking and Acceptance:

- Track and accept customer orders directly from the dashboard, ensuring timely processing.

Product Management:

- Create, edit, and delete products as per stock availability, maintaining an updated inventory.

User Interaction Management:

- Respond to user comments on products, fostering engagement and customer satisfaction.

Business Analytics:

- Utilize analytics tools within the dashboard to track shop performance and optimize strategies.

By providing both users and shop owners with tailored access, the platform ensures a seamless and efficient ecommerce experience. Users enjoy a diverse selection of products and interactive features, while shop owners benefit from a streamlined dashboard for effective management and growth.

Backend:

APIs and Technologies:

User Authentication API:

- **Endpoint:** /api/auth
- **Functionality:** Handles user authentication using JWT (JSON Web Tokens).
- **Methods:**
  - POST /api/auth/signup: User registration and JWT generation.
  - POST /api/auth/login: User login and JWT generation.
  - GET /api/auth/logout: User logout (optional).

Product Management API:

- **Endpoint:** /api/products
- **Functionality:** Manages products, categories, and market-related functionalities.
- **Methods:**
  - GET /api/products: Retrieve products based on different criteria (e.g., market, category).
  - GET /api/products/categories: Retrieve product categories.
  - POST /api/products/comment: Add a comment to a product.
  - POST /api/products/wishlist: Add a product to the user's wishlist.
  - POST /api/products/cart: Add a product to the user's cart.

Market Search API:

- **Endpoint:** /api/markets
- **Functionality:** Enables users to search products by selecting a specific market.
- **Methods:**
  - GET /api/markets: Retrieve available markets for selection.

Order Management API:

- **Endpoint:** /api/orders
- **Functionality:** Manages user orders and provides order tracking.
- **Methods:**
  - POST /api/orders: Place an order.
  - GET /api/orders/:orderId: Retrieve order details.
  - PUT /api/orders/:orderId/accept: Accept an order (Shop owner access).

Vendor Dashboard API:

- **Endpoint:** /api/vendor
- **Functionality:** Provides shop owners with a dashboard for order management and product updates.
- **Methods:**
  - GET /api/vendor/dashboard: Retrieve shop-related analytics and order information.
  - PUT /api/vendor/products: Create, edit, or delete products.

Email Notification API:

- **Integration:** Nodemailer for sending emails.
- **Functionality:** Sends email notifications for order confirmations, password resets, etc.
- **Methods:**
  - Custom methods for sending different types of emails.

Notification API:

- **Integration:** React Toast for displaying frontend notifications.
- **Functionality:** Displays notifications for actions like adding to cart, wishlist, successful orders, etc.
- **Methods:**
  - Integration within frontend components.

Stack:

- **Frontend:**
  - React.js for building user interfaces.
  - Axios for making API requests from the frontend.
  - React Toast for displaying notifications.
- **Backend:**
  - Node.js as the backend runtime.
  - Express.js for building the API endpoints.
  - MongoDB as the database.
  - Mongoose for interacting with MongoDB.
  - JSON Web Tokens (JWT) for user authentication.
- **Authentication:**
  - JWT for secure user authentication.
- **Email Handling:**
  - Nodemailer for sending transactional emails.
- **Notification Handling:**

    - React Toast for displaying user notifications.
- **Package Management:**
    - npm for managing packages.