### **USER STORY: ECOM PLATFORM**

Introduction:

This user story outlines the functionality of an e-commerce platform, consisting of two modules: Ecom Web App for users, Admin Panel for administrators.

#### 1. **Login Module**

* Description:
  + The user encounters the login module upon entering the web app.
  + If not registered, the user must first register to login.
* Login Options:
  + Users can sign in using Google, GitHub, or Linkedin accounts.
  + Alternatively, users can log in with a username and password.
* Successful Login:
  + Upon successful login, users are redirected to the homepage.

#### 1. **Sign Up**

* Description:
  + The user encounters the sign module upon entering the web app.
  + If not registered, the user must first register to login.
* Sign Up Options:
  + Users can sign in using Google, GitHub, or Linkedin accounts.
  + Alternatively, users can log in with a username and password.
* Successful Login:
  + Upon successful sign, users are redirected to the log in page.

#### 2. **Homepage Module**

* Navbar:
  + Logo: Redirects to the homepage.
  + Search: Allows users to search for specific products.
  + Login Icon: Redirects to the login and register page.
  + Return and Orders Section: Displays user orders; includes a "Return" button for initiating returns.
  + Cart Icon: Redirects to the cart page.
* Sidebar:
  + Implements filter functionality for products based on category ,name prices, and ratings.

4 **User Profile**

* + Users profile pic will be displayed on it.
  + On profile pic click it will show change profile pic, orders, user can update it’s info.

3 **Shopping Cart Module (Cart Page)**

As a buyer, I want to efficiently manage my shopping cart. Therefore:

* Modify Product Quantity:
  + I can adjust the quantity of a product using intuitive controls:
    - Increase quantity: Clicking the "+" button next to the product.
    - Decrease quantity: Clicking the "-" button next to the product.
* View Cart:
  + I can view all my carted products with quantity and price.
* Remove Products from Cart:
  + I can remove products from the cart.

#### 4. **Payment Module**

Description:

As a user, I want a seamless checkout experience with options for online and cash on delivery payments.

Features:

* Buy Now Button:
  + Clicking the "Buy Now" button initiates the checkout process.
* Checkout Form:
  + Users are prompted to fill in a small form with their customer name and order location,bank details .
  + The form serves as a quick input step before proceeding to payment options.
* Payment Options:
  + After submitting the form, users are directed to the Payment Module.
  + Users are presented with two payment options: "Online Payment" and "Cash on Delivery (COD)."
* Online Payment:
  + If the user selects "Online Payment," they are redirected to Stripe for a secure online payment process.
  + The amount is dynamically calculated based on the order details.
* Cash on Delivery (COD):
  + Users can opt for "Cash on Delivery" if they prefer to pay upon receiving the products.
  + Choosing COD does not require an immediate online transaction, allowing flexibility in payment.

#### 4. **Billing Module**

* Billing Options:
  + Online Payment: Users can download a billing PDF.
  + Cash on Delivery: Bill handed over by the delivery person and also can download PDF also.

#### X 5. **Product Tracking Module**

* Location Tracking:
  + Percentage display indicating the delivery progress.
  + Users can track the status of their order.
  + The Buyer gets a flash message on his contact number when his delivery is out.

#### 

#### 6. **Return Module**

As a user, I want a straightforward process for returning products, ensuring convenience and timely returns.

Features:

* Return Button:
  + After the order has been delivered, users can easily initiate a return by clicking on the "Return Button."
  + The "Return Button" is prominently displayed in the user interface for clear visibility.
* Time Limitation:
* The "Return Button" is available for users to click and initiate a return only within a specific timeframe.
  + Users can initiate returns within 2 days from the date of delivery.
* Dynamic Visibility:
  + The "Return Button" is dynamically displayed based on the delivery status and the time elapsed since delivery.
  + After the 2-day return window has passed, the "Return Button" is no longer visible to the user.

#### 7. **Refund Module**

* The refund will be initiated after 7 days after the order is returned it will be managed by the admin panel
* The refund “**will be credited on to the bank details of the checkout page** ‘that was entered by the user
* The refund will be through  **‘ Online’**

### 

### **ADMIN DASHBOARD**

**Admin Login:**

* As an admin, I access the login page.
* The login page has a simple Input field for entering my credentials. Credential will be a secret key for login.
* It includes a "Login" button for authentication.

#### 

#### 

#### 

#### **Admin Dashboard**

* **Sidebar**:

#### **Product Management**

* Manage Products:
  + After successful login, I land on the admin dashboard.
  + I see a navigation menu with options, including "Product Management."
  + Clicking on "Product Management" opens a page where I can:
    - View existing products.
    - Add new products with a form.
    - Edit and modify product details.
    - Delete products as needed.

**Order Management**

* Manage Orders:
  + In the dashboard, I navigate to "Order Management."
  + A page displays a list of orders with details like order number, date, and status.
  + Clicking on an order shows its details, including the products, customer information, and order status.
  + I have options to process and fulfill orders.
  + Filter by order type (placed, canceled, returned, refund).

#### **Customer Management**

* Manage Customers:
  + Under "Customer Management," I can view a list of registered users.
  + Clicking on a user displays their order history and profile details.
  + I can manage refunds for specific users.

#### 

#### **Reports**

* View Sales Reports:
  + In the "Reports" section, I can access sales reports.
  + The reports page displays graphs and charts representing sales data

#### **Payment And Finance**

* View Transactions:
  + The "Payment And Finance" section allows me to view and manage transactions.
  + A table displays transaction details, including date, amount, and payment status.

TECH STACK FOR THE PROJECT

The technology stack for the described e-commerce platform with user and admin modules can be chosen based on various factors such as development preferences, scalability, and the specific features you want to implement. Here's a generic suggestion for the tech stack:

### **Frontend (E-commerce Web App)**

* **React**.js: A popular JavaScript library for building user interfaces. It provides a component-based architecture for a modular and scalable frontend.
* **Redux**: For state management, especially when dealing with complex application states and data flow.
* **React Router**: For handling navigation and routing within the React application.
* **Auth0:** For secure and global login and signup
* **Axios**: A promise-based HTTP client for making API requests to the backend.
* **Material-UI:** UI component libraries to enhance the overall look and feel of the application.

### **Backend:**

* **Node.js**: A JavaScript runtime for building scalable server-side applications.
* **Express.js**: A lightweight web application framework for Node.js to build the backend server.
* **Postgres:** A SQL database for storing product information, user data, and order details.
* **Sequelize** is a popular Node.js ORM (Object-Relational Mapping) library for working with relational databases.
* **JWT** (JSON Web Tokens): For secure authentication and authorization.
* **Stripe API**: To handle online payments securely.

### **Payment and Finance:**

* **Stripe:** As mentioned, Stripe can be used to handle online payment transactions securely.

### 

### **Admin Dashboard:**

* **React.j:s** For building a responsive and interactive admin dashboard.
* **Redux**: For state management in the admin dashboard.
* **React Router:** For navigation within the admin dashboard.
* **Chart.js or D3.js**: For creating graphs and charts to represent sales data in the reports section.

### **Additional Tools and Services:**

* Firebase or AWS S3: For storing and serving static assets such as images
* Git and GitHub: For version control and collaborative development.