## **E-commerce Application Project Report**

**CARTSY**

### **Project Overview**

This project is an **E-commerce application** designed to allow users to browse products, add them to a cart, place orders, and manage their accounts. The application also includes an admin dashboard for managing customers, products, and orders. The system is built with a MySQL database backend and leverages Sequelize ORM for data interaction. Below, we outline the functionalities and database tables that power the application.

### **Functionality**

#### **1. User Side**

* **Login**: Users can log in with their registered email and password to access their account and order history.
* **Register**: New users can create an account by providing a username, email, and password.
* **Products**: Users can view a catalog of products, including their names, descriptions, prices, and images.
* **Cart**: Users can add products to a cart, adjust quantities, and view the cart's total value.
* **Order**: After reviewing their cart, users can place an order by providing delivery address details. They can view the status of their orders (Pending, Paid, or Delivered).

#### **2. Admin Dashboard**

* **Manage Products**: The admin can add, edit, and delete products, including setting prices, descriptions, and stock availability.
* **Manage Orders**: The admin can view all orders placed by users and update their status (Pending, Paid, or Delivered).
* **Manage Customers**: The admin can view, edit, and delete customer accounts as needed.
* **Admin Login**: A separate login for administrators ensures only authorized personnel can access the dashboard.

#### **3. Additional Features**

* **User Authentication**: Secure authentication ensures user data integrity, with hashed passwords stored in the database.
* **Stock Management**: The system automatically deducts stock quantities when an order is placed, preventing over-selling.
* **Order Tracking**: Users can monitor their orders' status via the "My Orders" page.

### **Database Tables**

#### **1. users Table**

Stores information about registered users.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| user\_id | INT (PK) | Unique ID for each user |
| username | VARCHAR(255) | User's chosen username |
|  |  |  |
| password | VARCHAR(255) | In plain text |
| email | VARCHAR(255) | User's email address (unique) |
| created\_at | TIMESTAMP | Timestamp when the account was created |

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### **2. products Table**

Stores information about products available in the store.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| Product\_id | INT (PK) | Unique ID for each product |
| Product\_name | VARCHAR(255) | Name of the product |
| description | TEXT | Detailed description of product |
| price | FLOAT | Price of the product |
| image | VARCHAR(255) | Path to the product image |
| Stock | INT | Quantity available in stock |
| createdAt | DATETIME | Timestamp when the product was added |
| updatedAt | DATETIME | Timestamp of the last update |

**3. cart Table**

Tracks products added to a user's cart.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| cart\_id | INT (PK) | Unique ID for each cart entry |
| user\_id | INT (FK) | References users(user\_id) |
| product\_id | INT (FK) | References products(Product\_id) |
| quantity | INT | Quantity of the product in cart |

#### **4. orders Table**

Stores information about orders placed by users.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| order\_id | INT (PK) | Unique ID for each order |
| user\_id | INT (FK) | References users(user\_id) |
| total\_price | FLOAT | Total value of the order |
| status | ENUM | Current status of the order (Pending, Paid, Delivered) |
| created\_at | TIMESTAMP | Timestamp when the order was placed |
| address | VARCHAR(255) | Delivery address for the order |

#### 

#### 

#### **5. order\_items Table**

Tracks products included in each order.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Unique ID for each entry |
| order\_id | INT (FK) | References orders(order\_id) |
| product\_id | INT (FK) | References products(Product\_id) |
| quantity | INT | Quantity of the product ordered |

#### **6. admins Table**

Stores administrator credentials.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| admin\_id | INT (PK) | Unique ID for each admin |
| username | VARCHAR(255) | Admin's username |
| password | VARCHAR(255) | Admin's hashed password |
| email | VARCHAR(255) | Admin's email address |
| created\_at | TIMESTAMP | Timestamp when the admin was added |

### 

### 

### **Technologies Used**

* **Backend**: Node.js with Sequelize ORM
* **Database**: MySQL PHPmyAdmin
* **Frontend**: Reactjs (or another framework as needed)
* **Authentication**: JSON Web Tokens (JWT) for session handling
* **Admin Panel**: Reactjs.

### **Future Enhancements**

1. **Wishlist Feature**: Allow users to save products they wish to purchase later.
2. **Search and Filter**: Enable users to search for products and filter by categories, price range, or ratings.
3. **Analytics Dashboard for Admin**: Provide insights into sales, user activity, and product performance.
4. **Review System**: Let users rate and review products.