**DBMS PROJECT REPORT**

Project Title: Ecommerce Management

Team members :

N PRANAV KUMAR - PES1UG22CS367

NAMAN BACHAVAT – PES1UG22CS370

#### **1. Purpose of the Project**

The Amazon clone aims to replicate essential eCommerce functionalities, allowing users to browse, shop, and manage orders efficiently while providing vendors a platform to list and sell products.

#### **2. Scope of the Project**

* **In Scope**:
  + User registration, login, and logout.
  + Product browsing, searching, and filtering.
  + Shopping cart management.
  + Secure payment options.
* **Out of Scope**:
  + AI-based product recommendations.
  + Multi-currency and multi-language support.

#### **3. Detailed Description**

##### **Roles:**

1. **Admin**:

Manages users, products, and orders.

**2.Vendor**:

Adds and manages product listings.

**3. Customer**:

Browses products, adds to the cart, and places orders.

##### **Database Relationships:**

* **Users Table**: Stores user details (customers, admins, and vendors).
* **Products Table**: Contains product details such as name, description, price, and stock.
* **Orders Table**: Tracks customer orders, including product IDs and quantities.
* **Payments Table**: Records payment details and statuses.

#### **4. Functional Requirements**

1. **User Registration and Login**:
   1. PHP-based authentication system.
2. **Product Management**:
   1. Vendors can add, update, and delete products.
3. **Shopping Cart**:
   1. Users can add/remove items and view totals.
4. **Checkout and Payment**:
   1. Secure options with payment gateway integration.

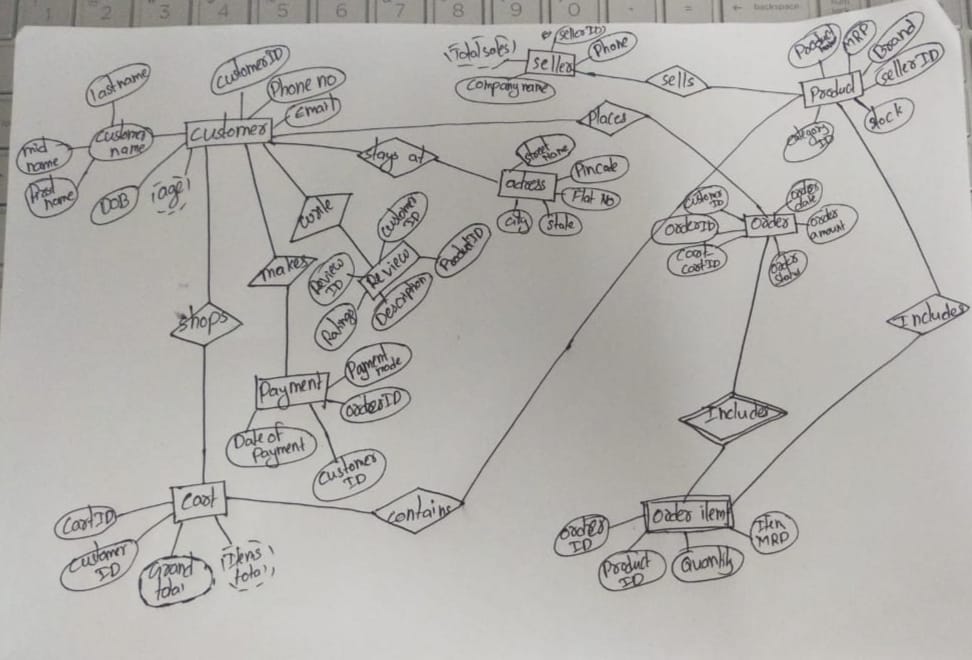
#### **Tools and Technologies**

**Frontend**: HTML, CSS, JavaScript.

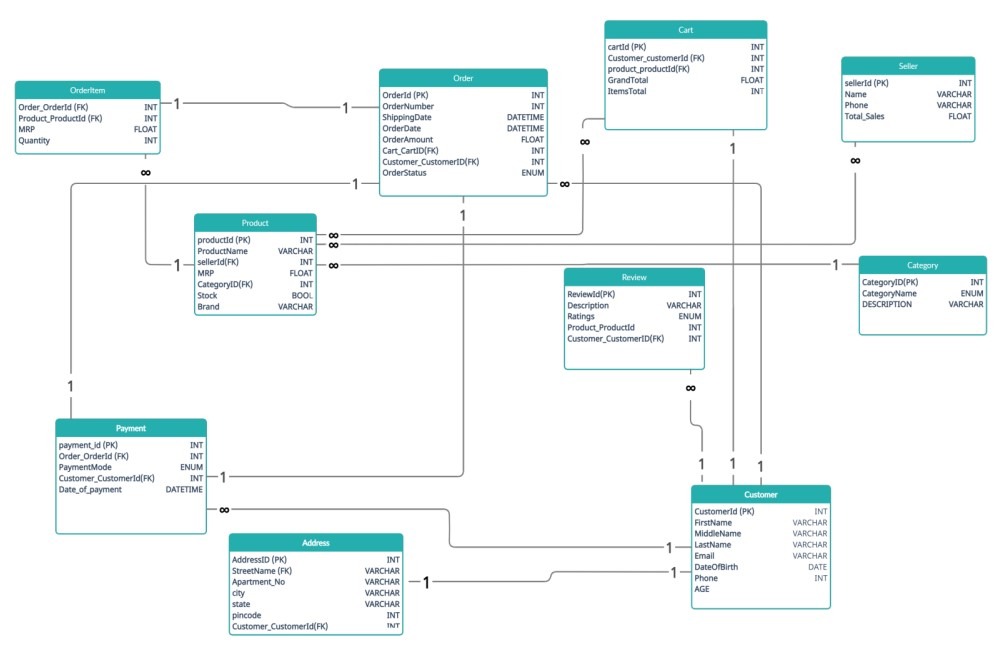
**Backend**: PHP with server-side logic.

**Database**: MySQL for data management.

#### **ER Diagram**



#### **Relational Schema**



DDL Commands:

--

-- Table structure for table `admins`

--

CREATE TABLE `admins` (

`admin\_id` int(100) NOT NULL,

`admin\_name` varchar(255) NOT NULL,

`admin\_email` varchar(255) NOT NULL,

`admin\_pass` varchar(255) NOT NULL,

`admin\_image` text NOT NULL,

`admin\_contact` varchar(255) NOT NULL,

`admin\_country` text NOT NULL,

`admin\_job` varchar(255) NOT NULL,

`admin\_about` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `admins`

--

INSERT INTO `admins` (`admin\_id`, `admin\_name`, `admin\_email`, `admin\_pass`, `admin\_image`, `admin\_contact`, `admin\_country`, `admin\_job`, `admin\_about`) VALUES

(5,’pranav', 'pranav123@gmail.com', '12345', '7015963438', 'india', 'I AM Pranav');

-- --------------------------------------------------------

--

-- Table structure for table `boxes\_section`

--

CREATE TABLE `boxes\_section` (

`box\_id` int(100) NOT NULL,

`box\_icon` varchar(100) NOT NULL,

`box\_title` varchar(255) NOT NULL,

`box\_desc` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `boxes\_section`

--

INSERT INTO `boxes\_section` (`box\_id`, `box\_icon`, `box\_title`, `box\_desc`) VALUES

(4, 'fa fa-trash', 'BEST IN MARKET', 'offer'),

(6, 'fa fa-shipping-fast fa-5', 'FAST SERVICE', 'raw'),

(7, 'fa fa-user', 'EDIT YOURSELF', 'edit'),

(8, 'fa fa-trash', 'DELETE EVERYTHING', 'delete');

-- --------------------------------------------------------

--

-- Table structure for table `cart`

--

CREATE TABLE `cart` (

`p\_id` int(100) NOT NULL,

`ip\_add` varchar(255) NOT NULL,

`qty` int(100) NOT NULL,

`size` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--

-- Table structure for table `categories`

--

CREATE TABLE `categories` (

`cat\_id` int(100) NOT NULL,

`cat\_title` text NOT NULL,

`cat\_desc` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `categories`

--

INSERT INTO `categories` (`cat\_id`, `cat\_title`, `cat\_desc`) VALUES

(7, 'SALOON PRODUCTS', 'All Saloon products'),

(8, 'PARLOR PRODUCTS', 'parlor'),

(9, 'GARMENTS', ''),

(10, 'DAILY-USE', ''),

(11, 'OTHERS', '');

-- --------------------------------------------------------

--

-- Table structure for table `customers`

--

CREATE TABLE `customers` (

`customer\_id` int(100) NOT NULL,

`customer\_name` varchar(255) NOT NULL,

`customer\_email` varchar(255) NOT NULL,

`customer\_pass` varchar(255) NOT NULL,

`customer\_country` text NOT NULL,

`customer\_city` text NOT NULL,

`customer\_contact` varchar(255) NOT NULL,

`customer\_address` text NOT NULL,

`customer\_image` text NOT NULL,

`customer\_ip` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `customers`

--

INSERT INTO `customers` (`customer\_id`, `customer\_name`, `customer\_email`, `customer\_pass`, `customer\_country`, `customer\_city`, `customer\_contact`, `customer\_address`, `customer\_image`, `customer\_ip`) VALUES

(23, 'raku', 'raku08@gmail.com', '12345', 'India', 'Bhiwani', '08295401008', 'Vpo- Alakhpura, Teh-Bawani khera', 'Licence.jpg', '::1'),

(24, 'pranav ', 'pranav123@gmail.com', '12345', 'India', 'Bhiwani', '+918295401008', 'Vpo- Alakhpura', 'IMG\_20180831\_153629.jpg', '::1'),

(25, 'Rakesh Kumar', 'rakekh@gmail.com', '12345', 'India', 'Bhiwani', '08295401008', 'Vpo- Alakhpura, Teh-Bawani khera', 'PHOTO.jpg', '::1'),

(26, 'jateen tanwar ', 'jateentanwar@gmail.com', '12345', 'India', 'Bhiwani', '08295401008', 'Vpo- Alakhpura, Teh-Bawani khera', 'Pic-4.jpg', '::1'),

(27, 'ankita', 'ankita@gmail.com', 'abcd', 'india', 'har', '1234567890', 'sampla', 'th.jpg', '::1'),

(28, 'Rakesh Kumar', 'rakeshalakh@gmail.com', '12345', 'India', 'Bhiwani', '08295401008', 'Vpo- Alakhpura, Teh-Bawani khera', 'youtube\_logo.png', '::1');

-- --------------------------------------------------------

--

-- Table structure for table `customer\_order`

--

CREATE TABLE `customer\_order` (

`order\_id` int(10) NOT NULL,

`customer\_id` int(10) NOT NULL,

`product\_id` int(100) NOT NULL,

`due\_amount` int(100) NOT NULL,

`invoice\_no` int(100) NOT NULL,

`qty` int(10) NOT NULL,

`size` text NOT NULL,

`order\_date` timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

`order\_status` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `customer\_order`

--

INSERT INTO `customer\_order` (`order\_id`, `customer\_id`, `product\_id`, `due\_amount`, `invoice\_no`, `qty`, `size`, `order\_date`, `order\_status`) VALUES

(32, 23, 18, 897, 1732273010, 3, 'RED & Blue', '2021-05-18 15:05:51', 'Complete'),

(33, 24, 17, 300, 2061043168, 1, 'RED & Blue', '2021-05-19 15:44:00', 'pending'),

(34, 26, 20, 425, 1601455995, 5, 'RED & Blue', '2021-05-25 16:41:16', 'Complete'),

(35, 26, 31, 2396, 1601455995, 4, 'RED & Blue', '2021-05-26 05:58:44', 'Complete'),

(36, 26, 33, 180, 1081656985, 1, 'RED & Blue', '2021-05-26 05:57:25', 'pending'),

(37, 27, 37, 10120, 675008297, 4, 'white', '2021-05-29 05:55:10', 'pending'),

(38, 28, 36, 599, 1624474385, 1, 'RED & Blue', '2021-06-19 10:25:24', 'Complete'),

(39, 28, 23, 349, 535526198, 1, 'RED & Blue', '2021-06-17 23:36:47', 'pending'),

(40, 28, 24, 432, 535526198, 1, 'RED & Blue', '2021-06-17 23:36:48', 'pending'),

(41, 28, 38, 1495, 437366899, 1, 'RED & Blue', '2021-06-19 10:24:50', 'pending');

-- --------------------------------------------------------

--

-- Table structure for table `payments`

--

CREATE TABLE `payments` (

`payment\_id` int(100) NOT NULL,

`invoice\_id` int(100) NOT NULL,

`amount` int(100) NOT NULL,

`payment\_mode` text NOT NULL,

`ref\_no` int(100) NOT NULL,

`payment\_date` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `payments`

--

INSERT INTO `payments` (`payment\_id`, `invoice\_id`, `amount`, `payment\_mode`, `ref\_no`, `payment\_date`) VALUES

(18, 236385455, 342, 'Paytm', 232323, '2021-05-18'),

(20, 1601455995, 599, 'google pay', 232323, '2021-05-11'),

(21, 1601455995, 599, 'Bank transfer', 112121, '2021-05-10'),

(22, 2098939645, 299, 'Bank transfer', 232323, '2021-06-17');

-- --------------------------------------------------------

--

-- Table structure for table `products`

--

CREATE TABLE `products` (

`product\_id` int(10) NOT NULL,

`p\_cat\_id` int(10) NOT NULL,

`cat\_id` int(10) NOT NULL,

`date` timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

`product\_title` text NOT NULL,

`product\_img1` text NOT NULL,

`product\_img2` text NOT NULL,

`product\_img3` text NOT NULL,

`product\_price` int(10) NOT NULL,

`product\_desc` text NOT NULL,

`product\_keyword` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `products`

--

INSERT INTO `products` (`product\_id`, `p\_cat\_id`, `cat\_id`, `date`, `product\_title`, `product\_img1`, `product\_img2`, `product\_img3`, `product\_price`, `product\_desc`, `product\_keyword`) VALUES

(21, 3, 5, '2021-05-20 08:35:26', 'Shaving Cream', 'crem.jpg', 'crem.jpg', 'crem.jpg', 299, 'erg', 'new arrival'),

(22, 14, 5, '2021-05-20 08:37:35', 'Towel', 'Pic-4.jpg', 'Pic-4.jpg', 'Pic-4.jpg', 199, 'best', 'Makhmal'),

(23, 14, 5, '2021-05-20 08:39:19', 'Hair Drayer', 's-l1600.jpg', 's-l1600.jpg', 's-l1600.jpg', 349, 'best', 'new arrival'),

(26, 1, 2, '2021-05-21 10:05:09', ‘pranav, 'garnier-Black-Naturalpacket-natural-Black-1.0.png', 'kemi.jpg', 'f6330a00-2dda-4c44-94d4-bfbf641d11a4.35a8aef2cceee4b1f480adc7148ba7a5.jpeg', 544, 'fg', 'new arrival lllllllllllll'),

(32, 10, 8, '2021-06-18 00:15:41', 'Stylish Comb', 'comb-11.jpg', 'com.jpg', 'th (1).jpg', 543, '', 'new arrival'),

(38, 21, 7, '2021-06-18 00:10:01', 'Philips trimmer for men', 'trimer1.jpg', 'trimer2.jpg', 'trimer3.jpg', 1495, '', 're');

-- --------------------------------------------------------

--

-- Table structure for table `product\_category`

--

CREATE TABLE `product\_category` (

`p\_cat\_id` int(10) NOT NULL,

`p\_cat\_title` text NOT NULL,

`p\_cat\_desc` text NOT NULL,

`p\_cat\_img` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `product\_category`

--

INSERT INTO `product\_category` (`p\_cat\_id`, `p\_cat\_title`, `p\_cat\_desc`, `p\_cat\_img`) VALUES

(22, 'TRIMMER', '', 0),

(23, 'DRAYER', '', 0),

(24, 'STRIGHTENER', '', 0),

(25, 'MASSAGER', '', 0),

(26, 'SHAVING CREAM', '', 0),

(27, 'BLADE', '', 0),

(28, 'NAPKIN', '', 0),

(29, 'LOION', '', 0),

(30, 'COLOR', '', 0),

(31, 'SCISSOR', '', 0),

(32, 'CLASSIC SHAVER', '', 0),

(33, 'COMB', '', 0),

(34, 'TOWEL', '', 0),

(35, 'SHAVING FOAM', '', 0),

(36, 'FACIAL KIT', '', 0),

(37, 'POWDER', '', 0),

(38, 'LIP CARE', '', 0),

(39, 'EYE LINER', '', 0),

(40, 'EYE LINER', '', 0),

(41, 'FACE CRAEM', '', 0),

(42, 'NAILPOLISH', '', 0),

(43, 'BEAUTICREAM', '', 0),

(44, 'LACME', '', 0),

(45, 'SCIN CARE', '', 0),

(46, 'INNER WEAR', '', 0),

(47, 'UNDER WEAR', '', 0),

(48, 'CAP', '', 0),

(49, 'HANKEY', '', 0),

(50, 'DETERGENT', '', 0),

(51, 'DEODODRANT', '', 0),

(52, 'FACEWASH', '', 0),

(53, 'HAIR GEL', '', 0),

(54, 'HARPIC', '', 0),

(55, 'PERFUME', '', 0),

(56, 'WALLET', '', 0),

(57, 'BELT', '', 0),

(58, 'SOAP', '', 0),

(59, 'TOOTHPASTE', '', 0),

(60, 'HANDWASH', '', 0),

(61, 'SHAMPOO', '', 0),

(62, 'HAIR OIL', '', 0),

(63, 'ROOM FRAGRANCE', '', 0),

(64, 'HAIR SPRAY', '', 0);

-- --------------------------------------------------------

--

-- Table structure for table `slider`

--

CREATE TABLE `slider` (

`Id` int(10) NOT NULL,

`slider\_name` varchar(255) NOT NULL,

`slider\_image` text NOT NULL,

`slider\_url` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `slider`

--

INSERT INTO `slider` (`Id`, `slider\_name`, `slider\_image`, `slider\_url`) VALUES

(18, 'sale', 'banner2.jpg', 'http://localhost/ecom/details.php?pro\_id=10'),

(23, '', 'banner5.png', ''),

(24, '', 'banner1.jpg', '');

--

-- Indexes for dumped tables

--

--

-- Indexes for table `admins`

--

ALTER TABLE `admins`

ADD PRIMARY KEY (`admin\_id`);

--

-- Indexes for table `boxes\_section`

--

ALTER TABLE `boxes\_section`

ADD PRIMARY KEY (`box\_id`);

--

-- Indexes for table `categories`

--

ALTER TABLE `categories`

ADD PRIMARY KEY (`cat\_id`);

--

-- Indexes for table `customers`

--

ALTER TABLE `customers`

ADD PRIMARY KEY (`customer\_id`);

--

-- Indexes for table `customer\_order`

--

ALTER TABLE `customer\_order`

ADD PRIMARY KEY (`order\_id`);

--

-- Indexes for table `payments`

--

ALTER TABLE `payments`

ADD PRIMARY KEY (`payment\_id`);

--

-- Indexes for table `products`

--

ALTER TABLE `products`

ADD PRIMARY KEY (`product\_id`);

--

-- Indexes for table `product\_category`

--

ALTER TABLE `product\_category`

ADD PRIMARY KEY (`p\_cat\_id`);

--

-- Indexes for table `slider`

--

ALTER TABLE `slider`

ADD PRIMARY KEY (`Id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `admins`

--

ALTER TABLE `admins`

MODIFY `admin\_id` int(100) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=6;

--

-- AUTO\_INCREMENT for table `boxes\_section`

--

ALTER TABLE `boxes\_section`

MODIFY `box\_id` int(100) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=9;

--

-- AUTO\_INCREMENT for table `categories`

--

ALTER TABLE `categories`

MODIFY `cat\_id` int(100) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=13;

--

-- AUTO\_INCREMENT for table `customers`

--

ALTER TABLE `customers`

MODIFY `customer\_id` int(100) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=29;

--

-- AUTO\_INCREMENT for table `customer\_order`

--

ALTER TABLE `customer\_order`

MODIFY `order\_id` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=42;

--

-- AUTO\_INCREMENT for table `payments`

--

ALTER TABLE `payments`

MODIFY `payment\_id` int(100) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=23;

--

-- AUTO\_INCREMENT for table `products`

--

ALTER TABLE `products`

MODIFY `product\_id` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=39;

--

-- AUTO\_INCREMENT for table `product\_category`

--

ALTER TABLE `product\_category`

MODIFY `p\_cat\_id` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=65;

--

-- AUTO\_INCREMENT for table `slider`

--

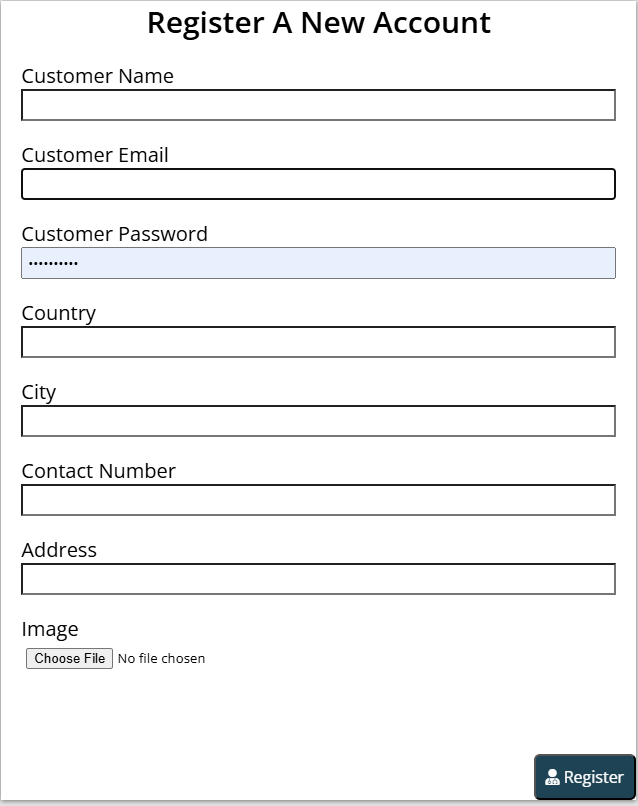
ALTER TABLE `slider`

MODIFY `Id` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=25;

COMMIT;

CRUD Operations:

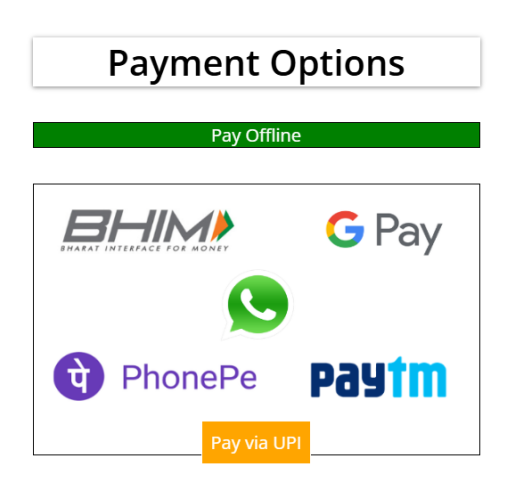
Register a new user:



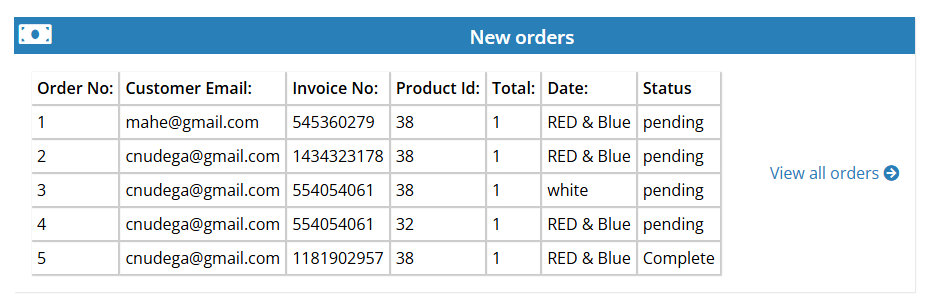
Cart:



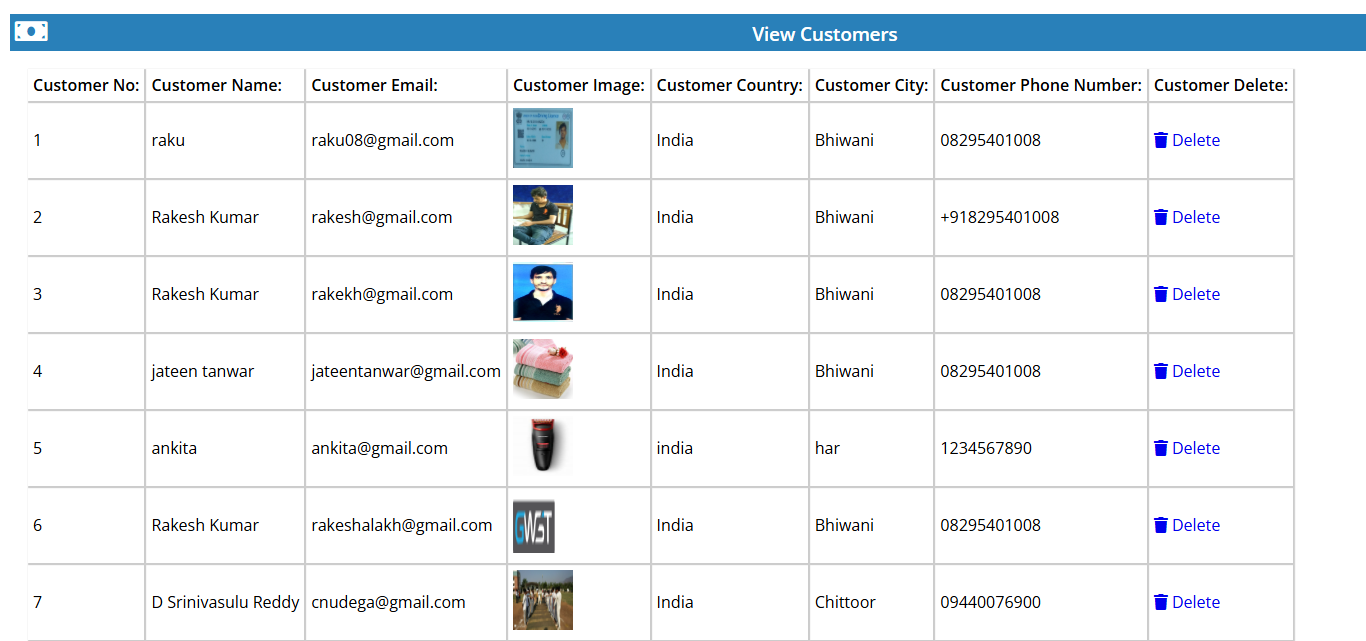
Proceeding to payment:



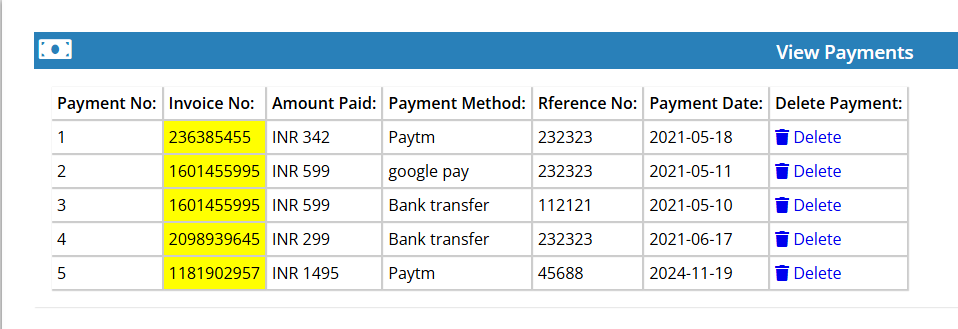
Orders placed by customer:



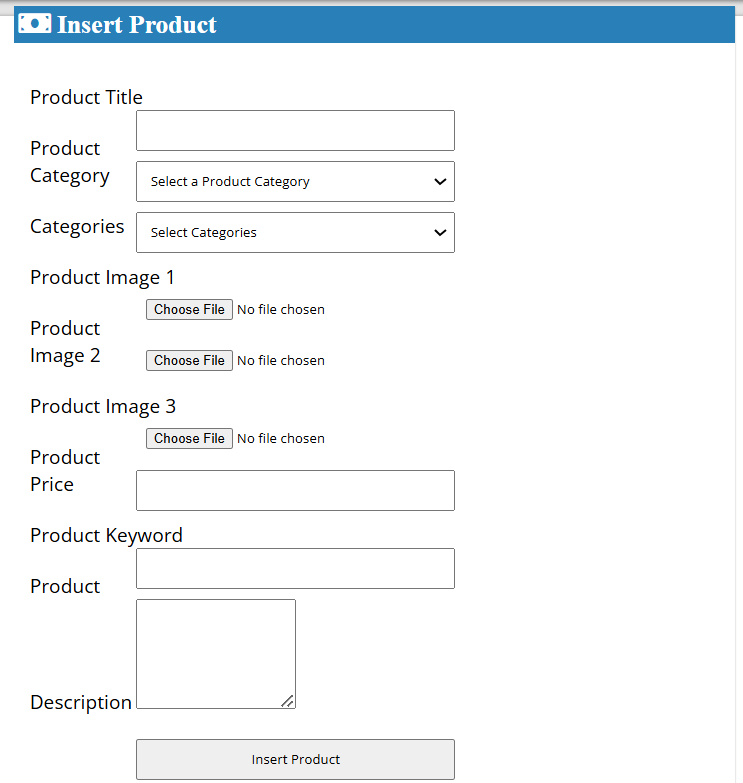
Customer Details:



Payments:



Inserting a Product:



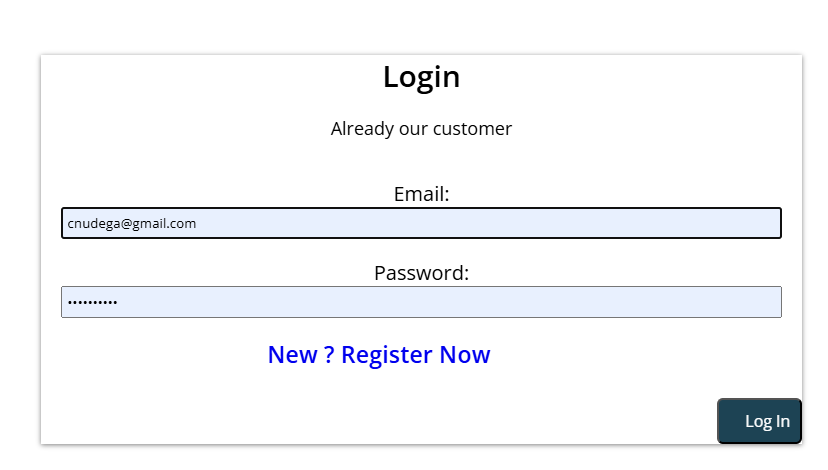
List of Functionalities:

· **Customer and Admin Registration and Login:** User-friendly registration and login interface.

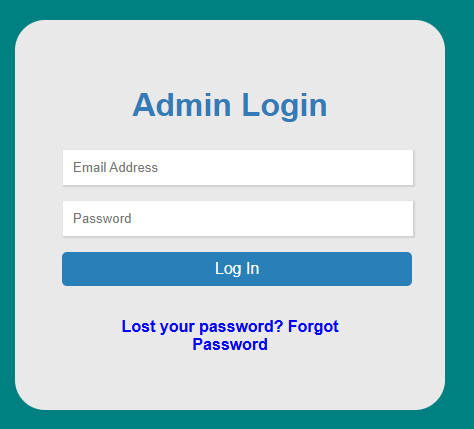
· **Customer Analysis in Admin Panel:** It displays total orders,total money spent,etc by each customer

· **Adding multiple items to cart and proceeding to payment:** Multiple items can be added to cart and different payment modes like offline pay,paytm,etc are provided

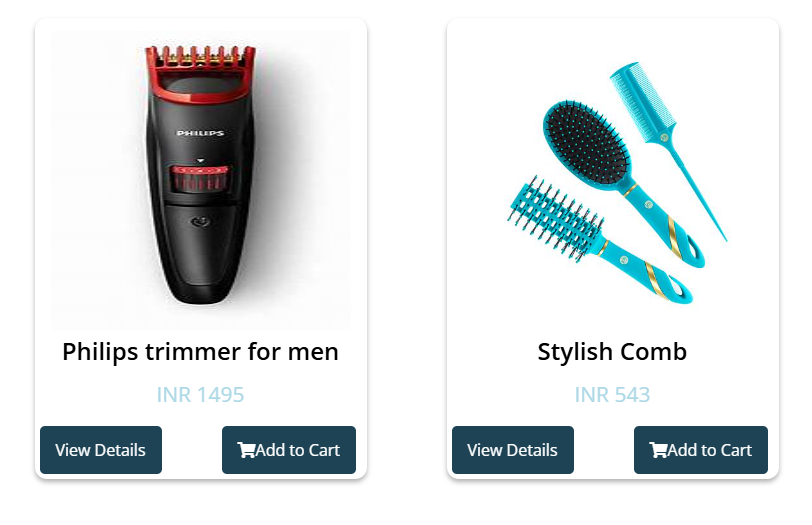
Login Page:



Admin login:



Adding to cart:



Customer Analysis:



Triggers,Procedures,Functions:

-- Update stored procedure without ALTER TABLE to avoid implicit commit

DELIMITER //

CREATE PROCEDURE update\_customer\_total\_orders(IN cust\_id INT)

BEGIN

    -- Update total orders for customer

    UPDATE customers

    SET total\_orders = (

        SELECT COUNT(\*)

        FROM customer\_order

        WHERE customer\_id = cust\_id

    )

    WHERE customer\_id = cust\_id;

END //

DELIMITER ;

-- Define the trigger without transaction control

DELIMITER //

CREATE TRIGGER after\_order\_insert

AFTER INSERT ON customer\_order

FOR EACH ROW

BEGIN

    CALL update\_customer\_total\_orders(NEW.customer\_id);

END//

DELIMITER ;

-- Function to calculate customer total spending

DELIMITER //

CREATE FUNCTION calculate\_customer\_total(cust\_id INT)

RETURNS DECIMAL(10,2)

DETERMINISTIC

READS SQL DATA

BEGIN

    DECLARE total DECIMAL(10,2);

    SELECT SUM(due\_amount) INTO total

    FROM customer\_order

    WHERE customer\_id = cust\_id

    AND order\_status = 'Complete';

    RETURN IFNULL(total, 0);

END //

DELIMITER ;

-- Stored procedure to get customer order history with product details

DELIMITER //

CREATE PROCEDURE get\_customer\_orders(IN cust\_id INT)

BEGIN

    SELECT

        co.order\_id,

        co.invoice\_no,

        p.product\_title,

        co.qty,

        co.due\_amount,

        co.order\_date,

        co.order\_status

    FROM customer\_order co

    JOIN products p ON co.product\_id = p.product\_id

    WHERE co.customer\_id = cust\_id

    ORDER BY co.order\_date DESC;

END //

DELIMITER ;

Code snippets for invoking:

-- Example 1: Call the update\_customer\_total\_orders procedure

CALL update\_customer\_total\_orders(1001);

-- Example 2: The trigger will automatically fire after inserting a new order

INSERT INTO customer\_order (

customer\_id,

product\_id,

invoice\_no,

qty,

due\_amount,

order\_status,

order\_date

) VALUES (

1001, -- customer\_id

5001, -- product\_id

'INV-2024-001', -- invoice\_no

2, -- qty

199.99, -- due\_amount

'Pending', -- order\_status

CURRENT\_TIMESTAMP -- order\_date

);

-- Example 3: Calculate customer's total spending using the function

-- Method 1: Direct SELECT

SELECT calculate\_customer\_total(1001) AS total\_spending;

-- Method 2: Use in a query

SELECT

customer\_id,

customer\_name,

calculate\_customer\_total(customer\_id) AS total\_spent

FROM customers

WHERE customer\_id = 1001;

-- Example 4: Get customer order history with product details

CALL get\_customer\_orders(1001);

-- Example 5: Using the function in a more complex query

SELECT

c.customer\_id,

c.customer\_name,

c.total\_orders,

calculate\_customer\_total(c.customer\_id) AS total\_spent,

CASE

WHEN calculate\_customer\_total(c.customer\_id) > 5000 THEN 'Premium'

WHEN calculate\_customer\_total(c.customer\_id) > 1000 THEN 'Regular'

ELSE 'New'

END AS customer\_segment

FROM customers c

WHERE c.customer\_id = 1001;

-- Example 6: Combining multiple operations in a transaction

START TRANSACTION;

-- Insert new order (trigger will fire automatically)

INSERT INTO customer\_order (

customer\_id, product\_id, invoice\_no, qty, due\_amount, order\_status, order\_date

) VALUES (

1001, 5001, 'INV-2024-002', 1, 99.99, 'Pending', CURRENT\_TIMESTAMP

);

-- Get updated totals

SELECT

customer\_id,

total\_orders,

calculate\_customer\_total(customer\_id) AS total\_spent

FROM customers

WHERE customer\_id = 1001;

-- View updated order history

CALL get\_customer\_orders(1001);

COMMIT;

Github Repo Link:

https://github.com/pranavnpk/ECOMMERCE-MANAGEMENT