

## **Final Project Guidelines System Integration & Architecture (CodeIgniter 4)**

Instructor: Zius D. Apresto

Student Name: Von Esson A. Vergara

Course: System Integration and Architecture

Level: 3rd Year (Individual Project)

Duration: 4 weeks

Platform: CodeIgniter 4 on XAMPP

## **1. System Overview:**

The Product Management System is a web-based application designed to streamline the management and monitoring of product information. It enables authorized users to add, edit, delete, and view products while automatically generating barcodes for easy identification. The system also provides user authentication, basic reporting features, and real-time currency conversion through an external API. Its primary purpose is to offer an organized and efficient platform for managing product records and supporting small-scale inventory operations.

## **2. System Architecture:**

### **Key Components**

#### **1. Controllers (app/Controllers/)**

Handle HTTP requests and business logic

- **AuthController.php** - Login, register, logout, session management
- **ProductController.php** - CRUD operations, API integration, currency conversion
- **PageController.php** - Dashboard, reports, product cards

#### **2. Models (app/Models/)**

Database interactions using Query Builder

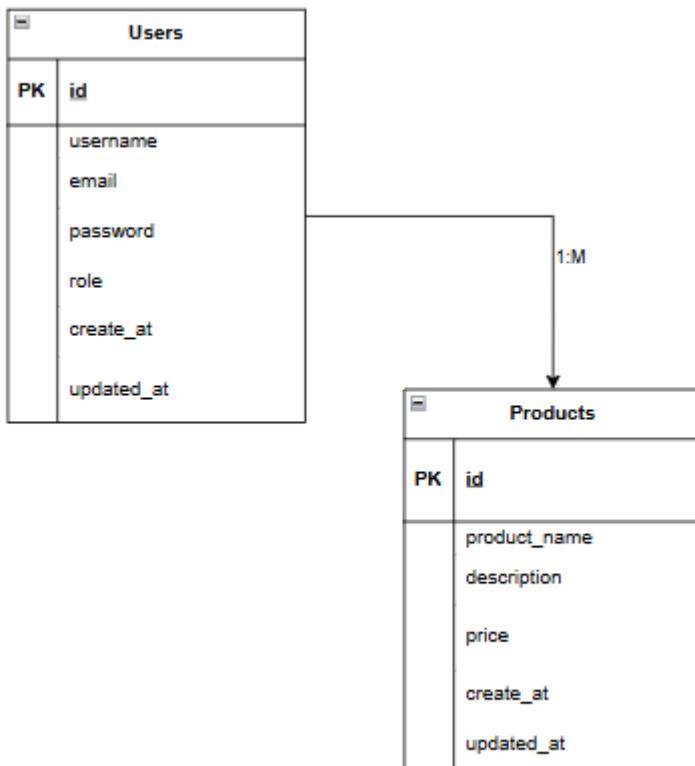
- **UserModel.php** - User authentication, role management
- **ProductModel.php** - Product CRUD with timestamps

#### **3. Views (app/Views/)**

HTML presentation layer

- **login.php** - Authentication form
- **register.php** - User registration
- **product\_view.php** - Main dashboard with CRUD
- **product\_card.php** - Product cards with barcodes
- **reports.php** – Analytics

### 3. Database Design:



1. users			
Column Name	Data Type	Constraints	Description
id	INT(11)	PK, AUTO_INCREMENT	Unique user identifier
username	VARCHAR(50)	UNIQUE, NOT NULL	User login name
email	VARCHAR(100)	UNIQUE, NOT NULL	User email address
password	VARCHAR(255)	NOT NULL	Bcrypt hashed password
role	ENUM	'admin', 'user'	User access level
created_at	DATETIME	NULL	Record creation timestamp
updated_at	DATETIME	NULL	Last modification timestamp

2. products			
Column Name	Data Type	Constraints	Description
id	INT(11)	PK, AUTO_INCREMENT	Unique product identifier
product_name	VARCHAR(255)	NOT NULL	Product name
description	TEXT	NULL	Product description (optional)
price	DECIMAL(10,2)	NOT NULL	Product price in USD
created_at	DATETIME	NULL	Record creation timestamp
updated_at	DATETIME	NULL	Last modification timestamp

## 🔗 Relationships

users ↔ products (Conceptual)

- **Type:** One-to-Many (1:N)
- **Description:** One user (admin) can manage multiple products
- **Implementation:** Currently implicit through session management
- **Note:** Foreign key not enforced; could be added as `user_id` in products table for audit trail

## 4. Authentication Module:

### Overview:

Simple session-based authentication system that allows users to securely login, manage their session, and logout. The system uses password hashing for security and session management for access control.

## **Login Process:**

The authentication module verifies user access through a username and password check using `password_verify()` against stored hashed credentials. When login is successful, a session is created to authorize the user for all system functions. Logout destroys the active session to prevent unauthorized reuse. The system does not include role management; all authenticated users have the same access level.

## **5. API Integration:**

### **API Provider: ExchangeRate-API**

**Type:** External REST API

**Endpoint:** <https://api.exchangerate-api.com/v4/latest/USD>

**Method:** GET

**Authentication:** None required (free tier)

#### **Purpose:**

Provides real-time foreign exchange rates to convert product prices from USD to multiple currencies, enabling international pricing display for global customers.

#### **Supported Currencies:**

- USD (US Dollar) - Base currency
- PHP (Philippine Peso)
- EUR (Euro)
- GBP (British Pound)
- JPY (Japanese Yen)
- AUD (Australian Dollar)
- CAD (Canadian Dollar)

#### **How It Works:**

1. System sends GET request to API endpoint
2. API returns JSON with current exchange rates for all currencies
3. Rates are cached for 1 hour to minimize API calls
4. Product prices are multiplied by selected currency rate
5. Converted prices displayed with appropriate currency symbols

**Response Format:**

The API returns a JSON object containing exchange rates relative to USD, the base currency code, and the current date of the rates.

### **Implementation Strategy:**

- **Caching:** Rates stored in CodeIgniter cache for 60 minutes
- **Fallback:** Default static rates used if API is unavailable
- **Performance:** Single API call serves all products
- **User Experience:** Dropdown selector triggers instant conversion

## **2. Barcode/QR Code Generation**

**Library: Picqer/PHP-Barcode-Generator (via Composer)**

**Type:** PHP library package

**Installation:** composer require picqer/php-barcode-generator

**Namespace:** Picqer\Barcode\BarcodeGenerator

**Purpose:**

Generates visual barcodes and QR codes for each product to facilitate:

- Inventory tracking
- Quick product identification
- Mobile scanning capabilities
- Print-ready product labels

**How It Works:**

1. Library loaded via Composer autoloader
2. Product ID or details passed to barcode generator
3. Barcode generated as PNG or SVG image
4. Image embedded in product card view or saved to storage
5. Can be printed for physical inventory labels

## **Features:**

- **Multiple Formats:** CODE\_128, EAN-13, QR Code, etc.
- **Customizable:** Size, color, and format options
- **Output Types:** PNG image, SVG vector, or HTML
- **No External API:** Generated locally on server
- **Lightweight:** Fast generation without network calls

## **Use Cases:**

- Product card displays with scannable barcodes
- Printable inventory labels
- Stock tracking and auditing
- Quick product lookup via barcode scanner
- Point-of-sale integration

## **3. Email Notification System**

### **Library: PHPMailer/PHPMailer (via Composer)**

**Type:** PHP email library package

**Installation:** composer require phpmailer/phpmailer

**Namespace:** PHPMailer\PHPMailer\PHPMailer

**Protocol:** SMTP (Simple Mail Transfer Protocol)

### **Purpose:**

Sends automated email notifications for:

- User registration confirmation
- Password reset requests
- Product inventory alerts
- System notifications
- Order confirmations (future enhancement)

### **Configuration Requirements:**

- SMTP server host and port (e.g., Gmail: smtp.gmail.com:587)

- Email credentials (username/password or app password)
- Sender email address and name
- TLS/SSL encryption settings

## **How It Works:**

1. PHPMailer library loaded via Composer autoloader
2. Event triggers email requirement (e.g., new user registration)
3. PHPMailer object configured with SMTP settings
4. Email composed with HTML template and content
5. Connects to SMTP server with credentials
6. Sends formatted email to recipient
7. Returns success/failure status

## **Features:**

- **HTML Emails:** Rich formatting with custom templates
- **Attachments:** Can send files (PDFs, images, etc.)
- **Multiple Recipients:** TO, CC, BCC support
- **Error Handling:** Detailed error messages for debugging
- **Security:** Supports TLS/SSL encryption
- **No External Service:** Uses your own SMTP server

**Integration Points:**

- User registration welcome emails
- Password recovery functionality
- Product stock alerts to admins
- Notification emails for updates
- Export reports via email

## **6. System Features / Modules**

### **Core CRUD Functionalities**

The Product Management System provides complete Create, Read, Update, Delete operations for managing product inventory efficiently.

#### **CREATE - Add New Product**

##### **Purpose:**

Allow users to add new products to the inventory database.

##### **Features:**

- Modal-based form for adding products
- Required fields: Product Name, Price
- Optional field: Description
- Real-time client-side validation
- Server-side validation with detailed error messages
- Automatic timestamp generation (created\_at)

##### **Validation Rules:**

- Product Name: 3-100 characters, letters/numbers/spaces only
- Description: 10-500 characters (optional)
- Price: Decimal format, must be greater than 0, max 999,999.99

##### **User Flow:**

Click "Add Product" button → Fill form → Submit → Validation → Save to database → Success message → Refresh product list

## **READ - View Products**

### **Purpose:**

Display all products from inventory with search and pagination capabilities.

### **Features:**

#### **Product Listing:**

- Table format showing all product details
- Columns: Product Name, Description, Price, Created Date, Updated Date, Actions
- Responsive design for mobile/desktop
- Print-friendly table layout

#### **Search Functionality:**

- Search by product name or description
- Real-time filtering
- Display result count
- Clear search option
- Maintains search across pagination

#### **Pagination:**

- 5 products per page
- Navigation controls (Previous/Next)
- Page numbers display
- Total product count
- Optimized database queries

#### **Dashboard Statistics:**

- Total product count
- Products created today
- Status indicators
- Overview cards with metrics

#### **Currency Display:**

- Prices shown in selected currency
- 7 currency options (USD, PHP, EUR, GBP, JPY, AUD, CAD)
- Currency symbols automatically displayed
- Real-time conversion from API

## **UPDATE - Edit Product**

### **Purpose:**

Modify existing product information while maintaining data integrity.

### **Features:**

- Edit button for each product row
- Modal pre-filled with current product data
- Same validation rules as Create
- Automatic timestamp update (updated\_at)
- Confirmation before saving changes

### **Editable Fields:**

- Product Name
- Description
- Price (in USD, converted for display)

### **User Flow:**

Click "Edit" → Modal opens with existing data → Modify fields → Submit → Validation → Update database → Success message → Refresh display

### **Data Preservation:**

- Original price maintained in USD
- Created date preserved
- Updated date automatically refreshed
- ID remains unchanged

## **DELETE - Remove Product**

### **Purpose:**

Safely remove products from inventory with confirmation.

### **Features:**

- Delete button for each product
- Confirmation modal to prevent accidental deletion
- Shows product name for verification
- Permanent removal from database
- Success notification after deletion

## **7. Security Implementation**

### **Security Measures**

The system implements multiple security layers to protect against common web vulnerabilities and ensure safe data handling.

### **Password Hashing**

**Algorithm:** Bcrypt (PASSWORD\_BCRYPT)

**Functions:** password\_hash() and password\_verify()

- Passwords never stored as plain text
- One-way encryption (cannot be reversed)
- Automatic salt generation per password
- Computationally expensive to crack

### **Process:**

- Registration: Hash password before storing in database
- Login: Compare entered password with stored hash using password\_verify()

## **CSRF Protection**

**Purpose:** Prevents Cross-Site Request Forgery attacks

- Unique token generated for each session
- Token included in all forms via `csrf_field()`
- Automatically validated on POST/PUT/DELETE requests
- Request rejected if token missing or invalid
- Enabled globally in CodeIgniter filters

**Protected:** Login, registration, product create/update/delete

## **SQL Injection Prevention**

**Method:** CodeIgniter Query Builder with prepared statements

- All database queries use parameter binding
- User input automatically escaped
- No raw SQL with user data
- Prevents malicious SQL code execution

## **XSS Prevention**

**Method:** Output escaping with `esc()` helper

- All user-generated content escaped before display
- Converts HTML/JavaScript to safe text
- Prevents script injection attacks
- Applied to product names, descriptions, usernames

## **Input Validation**

**Server-side validation on all inputs:**

**Product Fields:**

- Name: 3-100 chars, alphanumeric only
- Description: 10-500 chars (optional)
- Price: Decimal, 0.01 to 999,999.99

## **User Fields:**

- Username: Min 3 chars, unique
- Email: Valid format, unique
- Password: Min 6 chars

**Rejected data returns detailed error messages**

## **Session Security**

- Session data stored server-side
- Session ID regenerated on login (prevents fixation)
- Destroyed completely on logout
- HttpOnly cookies (JavaScript cannot access)
- Automatic timeout after inactivity
- Protected routes require valid session

## **Environment Configuration**

**Development:** Shows detailed errors for debugging

**Production:** Hides errors, logs to files only

## 8. Testing And Validation:

### Login:

The screenshot shows a clean, modern login interface. At the top center is the title "Login". Below it is a "Username" field containing "student". Underneath is a "Password" field with a visible password placeholder ".....". A "Remember me" checkbox is present. A large "Login" button is at the bottom. Below the button are links for "Forgot password?" and "No Account? [Register](#)". The entire form is contained within a white rounded rectangle.

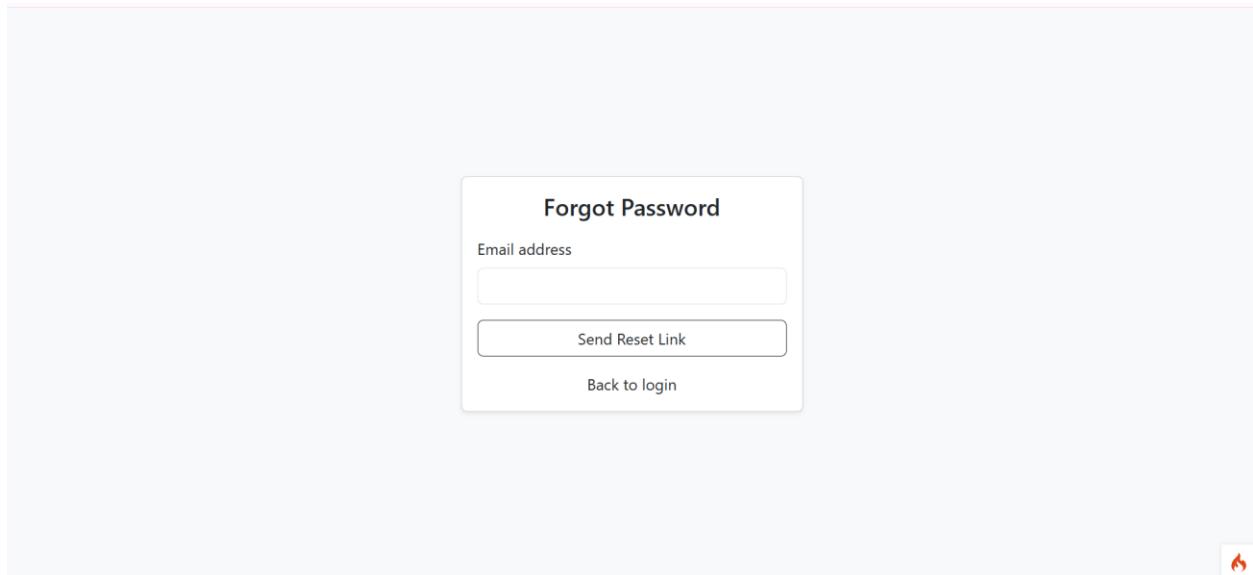
Login Page made with Bootstrap Framework with bootstrap icons making it more beautiful.

### Register:

The screenshot shows a registration form titled "Register". It includes fields for "Email" (placeholder "Email"), "Username" (placeholder "student"), and "Password" (placeholder "....."). A "Register" button is at the bottom. The form is styled similarly to the login page, with bootstrap icons and a clean design.

Allows the user to register, also made with bootstrap.

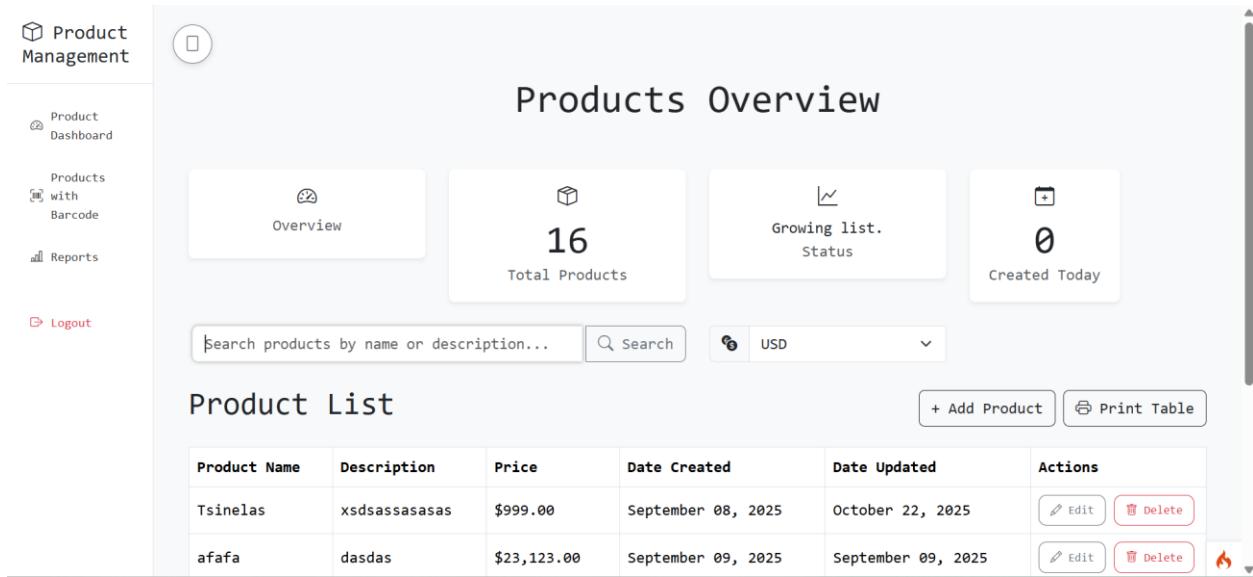
## Forgot Password:



A screenshot of a web-based forgot password form. The form is titled "Forgot Password" and contains a single input field labeled "Email address". Below the input field is a button labeled "Send Reset Link". At the bottom of the form is a link "Back to login". The background of the page is light gray.

Allows the user to change its password via gmail, this is integrated using Phpmailer a library of composer that allows the user to receive token to change password.

## User Dashboard:



A screenshot of a user dashboard for "Product Management". The sidebar on the left includes links for "Product Management", "Product Dashboard", "Products with Barcode", "Reports", and "Logout". The main area is titled "Products Overview" and features four cards: "Overview" (12 icons), "16 Total Products", "Growing list. Status" (1 icon), and "0 Created Today". Below this is a search bar with placeholder text "Search products by name or description..." and a currency selector set to "USD". The main content area is titled "Product List" and shows a table with two rows of data. The table columns are "Product Name", "Description", "Price", "Date Created", "Date Updated", and "Actions". The first row has a product named "Tsinelas" with a description of "xsdsassasasas" and a price of "\$999.00". The second row has a product named "afafa" with a description of "dasdas" and a price of "\$23,123.00". Each row in the table has edit and delete buttons in the "Actions" column. A small red fire icon is visible in the bottom right corner of the dashboard area.

Product Name	Description	Price	Date Created	Date Updated	Actions
Tsinelas	xsdssassasasas	\$999.00	September 08, 2025	October 22, 2025	<button>Edit</button> <button>Delete</button>
afafa	dasdas	\$23,123.00	September 09, 2025	September 09, 2025	<button>Edit</button> <button>Delete</button>

This part is where the CRUD in CI4 is done, with some enhancement like using currency converter API, cards with modals.

## Product Cards Section:

The screenshot shows a dashboard titled "Products Card With Barcode". On the left sidebar, there are links for "Product Management", "Product Dashboard", "Products with Barcode", "Reports", and "Logout". The main area displays a grid of six product cards. Each card contains a product name, a short description, a price, a barcode, and a unique identifier (e.g., 25, 26, 27). Below each card is a "View" button and a timestamp indicating when it was created. A small fire icon is located in the bottom right corner of the main area.

Product Name	Description	Price	Barcode	Identifier
Tsinelas	xsdssassasas	999.00		25
afafa	dasdas	23123.00		26
sdasdas	dsadas	312321.00		27
dasdads	dasdsad	23123.00		dasdads
sdasd	asdasd	12132.00		sdasd
asdasdjasjkj	dasdsadsdas	9999.00		asdasdjasjkj

This is the part where the barcode is generated automatically, this uses the library in the composer which is the vendor or the barcode generator.

## Reports:

The screenshot shows a dashboard titled "Reports & Analytics". On the left sidebar, there are links for "Product Management", "Product Dashboard", "Products with Barcode", "Reports", and "Logout". The main area displays three key metrics: "Total Products" (16), "Created Today" (0), and "Average Price" (₱45,172.74). Below these metrics is a line chart titled "Products Added (Last 6 months)". The chart shows a bell-shaped curve with data points at various intervals, starting near zero, peaking around the 6-month mark, and then declining. A small fire icon is located in the bottom right corner of the main area.

Metric	Value	Details
Total Products	16	All time
Created Today	0	2025-11-29
Average Price	₱45,172.74	Across listed products

This is the reports section, this is where we can see the status of our data, used chart.js for this. This is integrated by me, Mr. Von Esson Vergara.

## **9. System Setup Guide:**

### **Prerequisites**

- XAMPP (Apache + MySQL + PHP 7.4+)
- Composer
- Web Browser

### **Installation Steps**

Install CodeIgniter 4

```
composer create-project codeigniter4/appstarter SI-CRUD  
cd SI-CRUD
```

### **Install Dependencies**

```
composer require phpmailer/phpmailer
```

```
composer require picqer/php-barcode-generator
```

### **Configure Environment**

Copy env to [.env](#) and edit:

```
CI_ENVIRONMENT = development  
database.default.database = si_crud  
database.default.username = root  
database.default.password =
```

### **Create Database**

- Open phpMyAdmin: <http://localhost/phpmyadmin>
- Create database: si\_crud
- Import: [si-crud.sql](#)

### **Start XAMPP**

- Start Apache
- Start MySQL

### **Access Application**

<http://localhost/SI-CRUD/public/>

## **Default Login**

### **Register new account or use:**

- Username: admin
- Password: admin123

## **10. Conclusion / Future Improvements**

### **Project Summary**

The Product Management System successfully demonstrates a complete CRUD application built with CodeIgniter 4, integrating external APIs for currency conversion, barcode generation, and email capabilities. The system achieves all project requirements with secure authentication, database management, and a professional user interface.

### **Current Limitations**

1. No Image Upload - Products don't support photo uploads
2. Basic Reporting - Limited analytics and data visualization
3. Single Currency Storage - Prices stored only in USD
4. No Email Verification - Registration doesn't require email confirmation
5. Limited User Roles - Only basic user management (no role customization)
6. Manual Inventory - No automatic stock tracking or alerts
7. No Multi-language - Interface only in English
8. Local Deployment - Not configured for cloud/production hosting

### **Potential Enhancements**

#### **Short-term Improvements:**

1. Product Images - Upload and display photos
2. Categories - Organize products by category/type
3. Advanced Search - Filter by price range, date, category
4. CSV Export - Download product data for reports
5. Email Verification - Confirm email on registration
6. Stock Levels - Track inventory quantities

## **Long-term Improvements:**

1. Sales Module - Track transactions and revenue
2. Inventory Alerts - Low stock notifications
3. Multi-currency Payment - Accept payments in different currencies
4. REST API - Mobile app integration
5. Advanced Analytics - Charts, trends, forecasting
6. Barcode Scanner - Mobile scanning capability
7. Cloud Deployment - AWS/Azure hosting
8. Audit Trail - Log all user actions
9. Multi-language Support - Internationalization
10. Supplier Management - Track product suppliers

## **Reflection**

This project successfully demonstrates system integration principles by combining:

- MVC architecture for clean code organization
- External API integration for real-world data
- Security best practices for safe operation
- Database design for efficient data management

**VIEWS:****Login.php**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" integrity="sha384-tViUubb0FLlh3v/dWt3Jxw4gZQcNoSCxCIFNJVCx7D55/wXsrNIRANwdD" crossorigin="anonymous">
    <title>Login</title>
    <style>
        body {
            font-family: monospace;
        }
        .form-control:focus {
            box-shadow: none;
            outline: none;
            border-color: #aaa;
            /* optional: custom border color */
        }
    </style>
</head>
<body class="bg-light">
    <div class="d-flex justify-content-center align-items-center vh-100">
        <div class="card shadow-lg border-0 rounded-4" style="width: 320px; height: auto;">
            <div class="card-body px-4 py-5">
                <h4 class="card-title mb-4 fw-bold text-center"><i class="bi bi-person-check"></i>Login</h4>
                <form id="loginForm" action="= base_url('/login') ?" method="post" novalidate>
                    <!-- Username Field -->
                    <div class="mb-3">
                        <label for="username" class="form-label text-start w-100">Username</label>
                        <div class="input-group">
                            <span class="input-group-text"><i class="bi bi-person"></i></span>
                            <input type="text" class="form-control" id="username" name="username" placeholder="Enter your username" required>
                        </div>
                    </div>
                    <div class="mb-4">
                        <label for="password" class="form-label text-start w-100">Password</label>
                        <div class="input-group">
                            <span class="input-group-text"><i class="bi bi-lock"></i></span>
                            <input type="password" class="form-control" id="password" name="password" placeholder="Enter your password" required minlength="6">
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
```

```

        <button class="btn btn-outline-secondary" type="button" id="togglePassword" aria-label="Show password"><i class="bi bi-eye"></i></button>
    </div>
</div>

<div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="remember" name="remember">
    <label class="form-check-label small" for="remember">Remember me</label>
</div>

<div class="d-flex justify-content-center">
    <button type="submit" id="loginBtn" class="btn btn-outline-dark w-100">
        <span id="loginBtnSpinner" class="spinner-border spinner-border-sm me-2" role="status" style="display:none;" aria-hidden="true"></span>
        <span id="loginBtnText"><i class="bi bi-key"></i> Login</span>
    </button>
</div>
</form>

<!-- Optional Links -->
<div class="mt-4 text-center">
    <a href=<?= base_url('forgot-password') ?>" class="text-decoration-none small link-dark">Forgot password?</a><br>
    <a href="#" class="text-decoration-none small link-dark">No Account? <a href=<?php echo base_url('/register'); ?>" class="link-dark text-decoration-underline small">Register</a> </a>
</div>
</div>
</div>
</body>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
<!-- SweetAlert2 -->
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
<script>

    document.addEventListener('DOMContentLoaded', function() {
        <?php if (session()->getFlashdata('success')): ?>
            Swal.fire({
                icon: 'success',
                title: 'Registered Successfully',
                text: <?= json_encode(session()->getFlashdata('success')) ?>,
                width: 620,
                padding: '1em',
                confirmButtonText: 'OK'
            });
        <?php endif; ?>
        <?php if (session()->getFlashdata('error')): ?>
            Swal.fire({
                icon: 'error',
                title: 'Login Error',

```

```

text: <?= json_encode(session()->getFlashdata('error')) ?>,
confirmButtonText: 'OK'
});

<?php endif; ?>\

// Toggle password visibility (eye icon)

(function() {

var toggleBtn = document.getElementById('togglePassword');

var pwdInput = document.getElementById('password');

if (!toggleBtn || !pwdInput) return;

toggleBtn.addEventListener('click', function() {

var isHidden = pwdInput.type === 'password';

pwdInput.type = isHidden ? 'text' : 'password';

var icon = this.querySelector('i');

if (icon) {

icon.className = isHidden ? 'bi bi-eye-slash' : 'bi bi-eye';

}

this.setAttribute('aria-pressed', String(isHidden));

});

})();

});

</script>

</body>

</html>

```

### **Register.php**

```

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" integrity="sha384-tViUubbYAV00FLlhhi3v/dWt3Jxw4gZQcNoSCxCIFNJVCx7/D55/wXsrNIRANwdD" crossorigin="anonymous">

<link href="https://cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

<title>Register</title>

<style>

body {
    font-family: monospace;
}

```

```

.form-control:focus {
    box-shadow: none;
    outline: none;
    border-color: #aaa;
    /* optional: custom border color */
}

</style>

</head>

<body>

<div class="position-absolute" style="top:8px; left:8px; z-index:1050;">
    <a href="= base_url('/logout') ?" class="text-decoration-none text-dark align-items-center back-link" aria-label="Logout">
        <div class="card shadow-sm p-1" style="width:30px; height:30px; display:flex; align-items:center; justify-content:center;">
            <i class="bi bi-arrow-90deg-left fs-6"></i>
        </div>
    </a>
</div>

<div class="d-flex justify-content-center align-items-center vh-100">
    <div class="card shadow-lg border-0 rounded-4" style="width: 320px; height: auto;">
        <div class="card-body px-4 py-3">
            <strong class="text-center mb-0">
                <h3>Register</h3>
            </strong>
            <?php if (session()->getFlashdata('error')): ?>
                <div class="alert alert-danger"><?= session()->getFlashdata('error') ?></div>
            <?php endif; ?>
            <?php if (session()->getFlashdata('errors')): ?>
                <div class="alert alert-danger">
                    <ul class="mb-0">
                        <?php foreach (session()->getFlashdata('errors') as $err): ?>
                            <li><?= esc($err) ?></li>
                        <?php endforeach; ?>
                    </ul>
                </div>
            <?php endif; ?>
        <div id="registerForm" action="= base_url('/register') ?" method="post" class="mt-3 novalidate">
            <div class="mb-3">
                <label class="form-label small">Email</label>
                <div class="input-group">
                    <span class="input-group-text"><i class="bi bi-envelope"></i></span>

```

```

<input type="email" name="email" value=<?= esc(old('email')) ?>" class="form-control" placeholder="Email" required>
</div>
</div>
<div class="mb-3">
    <label class="form-label small">Username</label>
    <div class="input-group">
        <span class="input-group-text"><i class="bi bi-person"></i></span>
        <input type="text" name="username" value=<?= esc(old('username')) ?>" class="form-control" placeholder="Username" required>
    </div>
    </div>
    <div class="mb-3">
        <label class="form-label small">Password</label>
        <div class="input-group">
            <span class="input-group-text"><i class="bi bi-lock"></i></span>
            <input id="regPassword" type="password" name="password" class="form-control" placeholder="Password" required minlength="6">
            <button class="btn btn-outline-secondary" type="button" id="regTogglePassword"><i class="bi bi-eye"></i></button>
        </div>
        </div>
        <div class="d-grid">
            <button type="submit" id="registerBtn" class="btn btn-outline-dark w-100">
                <span id="registerBtnSpinner" class="spinner-border spinner-border-sm me-2" role="status" style="display:none;" aria-hidden="true"></span>
                <span id="registerBtnText">Register</span>
            </button>
        </div>
    </div>
</form>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
<!-- SweetAlert2 -->
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

<script>
    document.addEventListener('DOMContentLoaded', function() {
        // password toggle
        (function() {
            var t = document.getElementById('regTogglePassword');
            var p = document.getElementById('regPassword');
            if (!t || !p) return;
            t.addEventListener('click', function() {

```

```

var show = p.type === 'password';
p.type = show ? 'text' : 'password';
var ic = this.querySelector('i');
if (ic) ic.className = show ? 'bi bi-eye-slash' : 'bi bi-eye';
});

})();

// form submit spinner and basic validation

(function() {

var form = document.getElementById('registerForm');

var btn = document.getElementById('registerBtn');

var spinner = document.getElementById('registerBtnSpinner');

if (!form) return;

form.addEventListener('submit', function(e) {

if (!form.checkValidity()) {

e.preventDefault();

e.stopPropagation();

form.classList.add('was-validated');

return false;

}

if (spinner) spinner.style.display = "";

if (btn) btn.setAttribute('disabled', 'disabled');

return true;

}, false);

})();

// Flash messages (server-side)

<?php if (session()->getFlashdata('success')): ?>

Swal.fire({

icon: 'success',

title: 'Success',

text: <?= json_encode(session()->getFlashdata('success')) ?>,

confirmButtonText: 'OK'

});

<?php endif; ?>

<?php if (session()->getFlashdata('error')): ?>

Swal.fire({

icon: 'error',

title: 'Error',

text: <?= json_encode(session()->getFlashdata('error')) ?>,

confirmButtonText: 'OK'

});

```

```

        <?php endif; ?>
    });
</script>
</body>
</html>

Product_View.php
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Product List</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
</head>
<style>
body {
    font-family: monospace;
}

@media print {
    body * {
        visibility: hidden;
    }

    table,
    table *,
    .print {
        visibility: visible;
        /* show table and elements with class "print" */
    }
}

table {
    position: absolute;
    left: 0;
    top: 40px;
}

```

```

/* push it down a bit so the heading shows above */
width: 100%;

/* full width when printing */
}

.print {
position: absolute;
top: 0;
left: 0;
width: 100%;
text-align: absolute;
font-size: 20px;
font-weight: bold;
margin-bottom: 10px;
}

.no-print {
display: none !important;
}

}

.form-control:focus{box-shadow:0 0 4px rgba(0,0,0,0.4);border-color:#fff;transform:scale(0.98)}


.sidebar{width:150px;min-height:100vh;transition:all .3s;background:#fff;border-right:1px solid #ddd;position:fixed;top:0;left:0;z-index:1000}
.main-content{margin-left:150px;transition:all .3s;position:relative}
.toggle-btn{position:fixed;top:20px;z-index:2001;border-radius:50%;width:40px;height:40px;display:flex;align-items:center;justify-content:center;transition:left .25s,transform .25s;box-shadow:0 2px 6px rgba(0,0,0,.12);background:#fff;left:170px}
.sidebar.collapsed{margin-left:-250px}
.main-content.expanded{margin-left:0}
.sidebar.collapsed+.main-content .toggle-btn{left:12px}
.sidebar.collapsed+.main-content .toggle-btn i{transform:rotate(180deg)}
@media (max-width:768px){.sidebar{position:fixed;height:100vh}.main-content{margin-left:0}}
.nav-link.d-flex.align-items-center{font-size:.8rem;font-weight:500;color:#333}
.nav-link.d-flex.align-items-center:hover{background:#f8f9fa;color:#000;text-decoration:none}
.logout-pointer{cursor:pointer}
.btn-xs{padding:.2rem .7rem;font-size:.75rem;line-height:2;border-radius:.5rem}

</style>

<body class="bg-light">
<!-- ✅ SweetAlert2 for flash messages -->

```

```

<?php if (session()->getFlashdata('success')): ?>

<script>
Swal.fire({
  icon: 'success',
  title: '<?= session()->getFlashdata('success') ?>',
  width: 350,
  padding: '1.25rem',
  showConfirmButton: false,
  timer: 1600
});
</script>

<?php endif; ?>

<div class="d-flex">
  <!-- Sidebar -->
  <nav id="sidebar" class="sidebar bg-white border-end">
    <div class="sidebar-header d-flex align-items-center px-3 py-3 border-bottom">
      <h5 class="mb-0"><i class="bi bi-box"></i> Product Management</h5>
    </div>
    <ul class="nav flex-column px-2 mt-3">
      <li class="nav-item mb-2">
        <a class="nav-link d-flex align-items-center" href="<?= base_url('/dashboard') ?>">
          <i class="bi bi-speedometer2 me-2"></i> Product Dashboard
        </a>
      </li>
      <li class="nav-item mb-2">
        <a class="nav-link d-flex align-items-center" href="<?= base_url('/product_card') ?>">
          <i class="bi bi-upc-scan me-2"></i> Products with Barcode
        </a>
      </li>
      <li class="nav-item mb-2">
        <a class="nav-link d-flex align-items-center" href="<?= base_url('/reports') ?>">
          <i class="bi bi-bar-chart-line me-2"></i> Reports
        </a>
      </li>
      <li class="nav-item mt-4">
        <a href="<?= base_url('/logout') ?>" class="nav-link text-danger d-flex align-items-center logout-pointer logout-link">
          <i class="bi bi-box-arrow-right me-2"></i> Logout
        </a>
      </li>
    </ul>
  </nav>
</div>

```

```

</ul>
</nav>
<!-- Main Content --&gt;
&lt;div class="main-content flex-grow-1 p-4"&gt;
    <!-- Toggle button OUTSIDE the sidebar --&gt;
    &lt;button id="toggleSidebar" class="btn btn-outline-secondary toggle-btn"&gt;
        &lt;i class="bi bi-file"&gt;&lt;/i&gt;
    &lt;/button&gt;

    &lt;h1 class="text-center mt-5"&gt;Products Overview&lt;/h1&gt;
    &lt;div class="container mt-5"&gt;
        &lt;div class="row mb-4 text-center"&gt;
            <!-- Card 1: Overview --&gt;
            &lt;div class="col-md-3"&gt;
                &lt;div class="card shadow-sm border-0"&gt;
                    &lt;div class="card-body"&gt;
                        &lt;h5 class="card-title mb-2"&gt;&lt;i class="bi bi-speedometer2"&gt;&lt;/i&gt;&lt;/h5&gt;
                        &lt;h6 class="text-muted"&gt;Overview&lt;/h6&gt;
                    &lt;/div&gt;
                &lt;/div&gt;
            &lt;/div&gt;
            <!-- Card 2: Total Products --&gt;
            &lt;div class="col-md-3"&gt;
                &lt;div class="card shadow-sm border-0"&gt;
                    &lt;div class="card-body"&gt;
                        &lt;h5 class="card-title mb-2"&gt;&lt;i class="bi bi-box-seam"&gt;&lt;/i&gt;&lt;/h5&gt;
                        &lt;h1 class="display-6 mb-0 text-dark"&gt;
                            &lt;?= isset($totalCount) ? $totalCount : (isset($products) ? count($products) : 0) ?&gt;
                        &lt;/h1&gt;
                        &lt;span class="text-muted"&gt;Total Products&lt;/span&gt;
                    &lt;/div&gt;
                &lt;/div&gt;
            &lt;/div&gt;
            <!-- Card 3: Status --&gt;
            &lt;div class="col-md-3"&gt;
                &lt;div class="card shadow-sm border-0"&gt;
                    &lt;div class="card-body"&gt;
                        &lt;h5 class="card-title mb-2"&gt;&lt;i class="bi bi-graph-up"&gt;&lt;/i&gt;&lt;/h5&gt;
                        &lt;p class="mb-0 fs-6"&gt;
                            &lt;?php
</pre>

```

```

$count = isset($totalCount) ? $totalCount : (isset($products) ? count($products) : 0);

if ($count === 0) {
    echo "No products yet.";
} elseif ($count < 5) {
    echo "Few products <i class='bi bi-graph-down'></i>";
} elseif ($count < 20) {
    echo "Growing list.";
} else {
    echo "Large inventory!";
}

?>

</p>
<span class="text-muted">Status</span>

</div>
</div>
</div>

<!-- Card 4: Products Created Today --&gt;

&lt;div class="col-md-2"&gt;
    &lt;div class="card shadow-sm border-0"&gt;
        &lt;div class="card-body"&gt;
            &lt;h5 class="card-title mb-2"&gt;&lt;i class="bi bi-calendar-plus" style="font-size: 1.5em; color: #007bff;"&gt;&lt;/i&gt;&lt;/h5&gt;
            &lt;h1 class="display-6 mb-0 text-dark"&gt;
                &lt;?php
                    $today = date('Y-m-d');
                    $createdToday = 0;
                    // Get all products created today from the database, not just current page
                    try {
                        $productModel = new \App\Models\ProductModel();
                        $todaysProducts = $productModel-&gt;where('DATE(created_at)', $today)-&gt;findAll();
                        $createdToday = count($todaysProducts);
                    } catch (Exception $e) {
                        // Fallback to current page products if database query fails
                        if (!empty($products)) {
                            foreach ($products as $p) {
                                if (date('Y-m-d', strtotime($p['created_at'])) === $today) {
                                    $createdToday++;
                                }
                            }
                        }
                    }
                &lt;/?php&gt;
            &lt;h1&gt;{$createdToday}&lt;/h1&gt;
            &lt;p&gt;{$today}&lt;/p&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre>

```

```

        }

        echo $createdToday;

    ?>

</h1>

<span class="text-muted">Created Today</span>

</div>

</div>

</div>

</div>

<!-- Search Bar and Currency Selector -->

<div class="row mb-4">

<div class="col-md-6">

<form method="GET" action="= base_url('/dashboard') ?&gt;" class="d-flex"&gt;

&lt;div class="input-group"&gt;

&lt;input type="text" name="keyword" class="form-control"

placeholder="Search products by name or description..." 

value="<?= isset($keyword) ? esc($keyword) : 'USD' ?&gt;"&gt;

&lt;input type="hidden" name="currency" value="<?= isset($selectedCurrency) ? esc($selectedCurrency) : 'USD' ?&gt;"&gt;

&lt;button class="btn btn-outline-secondary" type="submit"&gt;

&lt;i class="bi bi-search"&gt;&lt;/i&gt; Search

&lt;/button&gt;

&lt;?php if (isset($keyword) &amp;&amp; !empty($keyword)): ?&gt;

&lt;a href="<?= base_url('/dashboard?currency=' . (isset($selectedCurrency) ? esc($selectedCurrency) : 'USD')) ?&gt;" 

class="btn btn-outline-danger"&gt;

&lt;i class="bi bi-x-circle"&gt;&lt;/i&gt; Clear

&lt;/a&gt;

&lt;?php endif; ?&gt;

&lt;/div&gt;

&lt;/form&gt;

&lt;/div&gt;

&lt;div class="col-md-3"&gt;

&lt;form method="GET" action="<?= base_url('/dashboard') ?&gt;" id="currencyForm"&gt;

&lt;div class="input-group"&gt;

&lt;label class="input-group-text"&gt;&lt;i class="bi bi-currency-exchange"&gt;&lt;/i&gt;&lt;/label&gt;

&lt;select name="currency" class="form-select" onchange="document.getElementById('currencyForm').submit()"&gt;

&lt;?php foreach ($availableCurrencies as $curr): ?&gt;

&lt;option value="<?= $curr ?&gt;" &lt;?= (isset($selectedCurrency) &amp;&amp; $selectedCurrency === $curr) ? 'selected' : '' ?&gt;&gt;

&lt;?= $curr ?&gt;

&lt;/option&gt;

&lt;?php endforeach; ?&gt;

&lt;/select&gt;
</pre

```

```

<input type="hidden" name="keyword" value="<?= isset($keyword) ? esc($keyword) : " ?>">
</div>
</form>
</div>
<div class="col-md-3">
    <?php if (isset($keyword) && !empty($keyword)): ?>
        <div class="alert alert-info mb-0 py-2">
            <small><i class="bi bi-info-circle"></i> Found <?= $totalCount ?> result(s)</small>
        </div>
    <?php endif; ?>
</div>
</div>
<div class="d-flex justify-content-between align-items-center mb-3">
    <h2 class="print">Product List</h2>
    <div class="d-flex gap-2">
        <button type="button" class="btn btn-outline-dark" data-bs-toggle="modal" data-bs-target="#addProductModal">
            + Add Product
        </button>
        <button class="btn btn-outline-dark" onclick="window.print()"><i class="bi bi-printer"></i> Print Table</button>
    </div>
</div>
<!-- Add Product Modal -->
<div class="modal fade" id="addProductModal" tabindex="-1" aria-labelledby="addProductModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header bg-light text-dark">
                <h5 class="modal-title" id="addProductModalLabel"><i class="bi bi-plus"></i> Add New Product</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <form action="<?= base_url('products/store') ?>" method="post" id="addProductForm" novalidate>
                    <div class="mb-3">
                        <label for="product_name" class="form-label">Product Name <span class="text-danger">*</span></label>
                        <input type="text" id="product_name" name="product_name" class="form-control" required minlength="3" maxlength="100" pattern="[A-Za-z0-9\s\-\_]+"
                            title="Product name must be 3-100 characters and contain only letters, numbers, spaces, hyphens, and underscores">
                    <div class="invalid-feedback">
                        Product name is required (3-100 characters, letters/numbers only)
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

```

```

        </div>

        <div class="mb-3">
            <label for="description" class="form-label">Description</label>
            <textarea id="description" name="description" class="form-control" rows="3"
                minlength="10" maxlength="500"
                placeholder="Enter product description (optional, min 10 characters if provided)"></textarea>
            <div class="invalid-feedback">
                Description must be between 10-500 characters if provided
            </div>
        </div>

        <div class="mb-3">
            <label for="price" class="form-label">Price (USD) <span class="text-danger">*</span></label>
            <input type="number" step="0.01" id="price" name="price" class="form-control"
                required min="0.01" max="999999.99"
                placeholder="0.00">
            <div class="invalid-feedback">
                Price is required and must be greater than 0
            </div>
        </div>

        <div class="d-flex justify-content-end">
            <button type="submit" class="btn btn-outline-dark">Save Product</button>
        </div>
    </form>
</div>
</div>
</div>

<table class="table table-bordered table-hover align-middle">
    <thead class="table">
        <tr>
            <th>Product Name</th>
            <th>Description</th>
            <th>Price</th>
            <th>Date Created</th>
            <th>Date Updated</th>
            <th width="180" class="no-print">Actions</th>
        </tr>
    </thead>
    <tbody>
        <?php if (!empty($products)): ?>

```

```

<?php foreach ($products as $p): ?>

<tr>
    <td><?= $p['product_name'] ?></td>
    <td><?= $p['description'] ?></td>
    <td><?= isset($p['currency_symbol']) ? $p['currency_symbol'] : '$' ?><?= number_format($p['price'], 2) ?></td>
    <td><?= date('F d, Y', strtotime($p['created_at'])) ?></td>
    <td><?= date('F d, Y', strtotime($p['updated_at'])) ?></td>
    <td>
        <!-- Edit Button -->
        <button type="button" class="btn btn-outline-secondary btn-xs no-print" data-bs-toggle="modal" data-bs-target="#editProductModal<?= $p['id'] ?>"><i class="bi bi-pencil"></i> Edit</button>
        <!-- Delete Button -->
        <button type="button" class="btn btn-outline-danger btn-xs no-print" data-bs-toggle="modal" data-bs-target="#deleteProductModal<?= $p['id'] ?>"><i class="bi bi-trash"></i> Delete</button>
    </td>
</tr>
<!-- Edit Product Modal -->
<div class="modal fade" id="editProductModal<?= $p['id'] ?>" tabindex="-1" aria-labelledby="editProductModalLabel<?= $p['id'] ?>" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header bg-secondary text-dark">
                <h5 class="modal-title text-light id=" editProductModalLabel<?= $p['id'] ?>">Edit Product</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <form action="<?= base_url('products/update/' . $p['id']) ?>" method="post" class="editProductForm" novalidate>
                    <div class="mb-3">
                        <label for="edit_product_name<?= $p['id'] ?>" class="form-label">Product Name <span class="text-danger">*</span></label>
                        <input type="text" id="edit_product_name<?= $p['id'] ?>" name="product_name" class="form-control" value="<?= esc($p['product_name']) ?>" required minlength="3" maxlength="100" pattern="[A-Za-z0-9\s\-\_]+" title="Product name must be 3-100 characters and contain only letters, numbers, spaces, hyphens, and underscores">
                    <div class="invalid-feedback">
                        Product name is required (3-100 characters, letters/numbers only)
                    </div>
                </div>
                <div class="mb-3">
                    <label for="edit_description<?= $p['id'] ?>" class="form-label">Description</label>
                    <textarea id="edit_description<?= $p['id'] ?>" name="description" class="form-control" rows="3">
                </div>
            </div>
        </div>
    </div>

```

```

minlength="10" maxlength="500"
placeholder="Enter product description (optional, min 10 characters if provided)"><?= esc($p['description'])

?></textarea>

<div class="invalid-feedback">
    Description must be between 10-500 characters if provided
</div>
</div>

<div class="mb-3">
    <label for="edit_price<?= $p['id'] ?>" class="form-label">Price (USD) <span class="text-danger">*</span></label>
    <input type="number" step="0.01" id="edit_price<?= $p['id'] ?>" name="price" class="form-control" value="<?= esc(isset($p['original_price']) ? $p['original_price'] : $p['price']) ?>" required min="0.01" max="999999.99">
    <div class="invalid-feedback">
        Price is required and must be greater than 0
    </div>
</div>
<div class="d-flex justify-content-end">
    <button type="submit" class="btn btn-outline-dark">Update Product</button>
</div>
</form>
</div>
</div>
<!-- Delete Product Modal -->
<div class="modal fade" id="deleteProductModal<?= $p['id'] ?>" tabindex="-1" aria-labelledby="deleteProductModalLabel<?= $p['id'] ?>" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header bg-danger text-white">
                <h5 class="modal-title" id="deleteProductModalLabel<?= $p['id'] ?>">Delete Product</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <p>Are you sure you want to delete <strong><?= $p['product_name'] ?></strong>?</p>
            </div>
            <div class="modal-footer">
                <form action="<?= base_url('products/delete/' . $p['id']) ?>" method="post">
                    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancel</button>
                    <button type="submit" class="btn btn-danger">Delete</button>
                </form>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
        </div>
        </div>
        </div>
        </tr>
    <?php endforeach; ?>
    <?php else: ?>
        <tr>
            <td colspan="5" class="text-center">No products found.</td>
        </tr>
    <?php endif; ?>
</tbody>
</table>

<!-- Bootstrap Pagination -->
<?php if (isset($pager) && $pager->getPageCount() > 1) : ?>
    <div class="d-flex justify-content-center mt-4">
        <?= $pager->links('default', 'bootstrap_pagination') ?>
    </div>
<?php endif; ?>

<!-- Pagination Info -->
<?php if (isset($pager)) : ?>
    <div class="d-flex justify-content-center mt-2">
        <small class="text-muted">
            <?php
                $details = $pager->getDetails();
                $currentPage = $details['currentPage'];
                $perPage = $details['perPage'];
                $total = $details['total'];
                $firstItem = (($currentPage - 1) * $perPage) + 1;
                $lastItem = min($currentPage * $perPage, $total);
            ?>
            Showing <?= $firstItem ?> to <?= $lastItem ?> of <?= $total ?> products
        </small>
    </div>
<?php endif; ?>
</div>

<!-- Flash Messages (SweetAlert2) -->

```

```

<?php if (session()->getFlashdata('message')): ?>

<script>
    document.addEventListener('DOMContentLoaded', function() {
        Swal.fire({
            icon: 'success',
            title: 'Success',
            text: "<?= esc(session()->getFlashdata('message')) ?>",
            confirmButtonText: 'OK'
        });
    });
</script>

<?php endif; ?>

<?php if (session()->getFlashdata('error')): ?>

<script>
    document.addEventListener('DOMContentLoaded', function() {
        Swal.fire({
            icon: 'error',
            title: 'Error',
            text: "<?= esc(session()->getFlashdata('error')) ?>",
            confirmButtonText: 'OK'
        });
    });
</script>

<?php endif; ?>

</div> <!-- End of main-content -->

</div> <!-- End of d-flex -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

<!-- Search Enhancement Script -->

<script>
    // Focus on search input when page loads if there's a search term
    document.addEventListener('DOMContentLoaded', function() {
        const searchInput = document.querySelector('input[name="keyword"]');
        const urlParams = new URLSearchParams(window.location.search);
        const keyword = urlParams.get('keyword');

        // Auto-focus search input if there's no search term

```

```

if (!keyword && searchInput) {
    searchInput.focus();
}

// Add search on Enter key
if (searchInput) {
    searchInput.addEventListener('keypress', function(e) {
        if (e.key === 'Enter') {
            e.preventDefault();
            this.closest('form').submit();
        }
    });
}

// Initialize form validation
initializeFormValidation();

});

// Add loading state to search button
document.querySelector('form').addEventListener('submit', function() {
    const searchBtn = this.querySelector('button[type="submit"]');
    if (searchBtn && searchBtn.type === 'submit' && this.querySelector('input[name="keyword"]')) {
        const originalText = searchBtn.innerHTML;
        searchBtn.innerHTML = '<i class="bi bi-hourglass-split"></i> Searching...';
        searchBtn.disabled = true;

        // Re-enable after a short delay (in case of errors)
        setTimeout(() => {
            searchBtn.innerHTML = originalText;
            searchBtn.disabled = false;
        }, 3000);
    }
});

// Form validation function
function initializeFormValidation() {
    // Add validation to all product forms
    const forms = document.querySelectorAll('#addProductForm, .editProductForm');

    forms.forEach(form => {
        form.addEventListener('submit', function(event) {

```

```

if (!form.checkValidity()) {
    event.preventDefault();
    event.stopPropagation();
}

// Custom validation for description (optional but min length if provided)
const description = form.querySelector('textarea[name="description"]');

if (description && description.value.trim() !== "" && description.value.trim().length < 10) {
    description.setCustomValidity('Description must be at least 10 characters if provided');
    event.preventDefault();
    event.stopPropagation();
} else if (description) {
    description.setCustomValidity("");
}

form.classList.add('was-validated');
});

// Real-time validation feedback
const inputs = form.querySelectorAll('input, textarea');
inputs.forEach(input => {
    input.addEventListener('input', function() {
        if (input.name === 'description') {
            // Special handling for optional description
            if (input.value.trim() === "" || input.value.trim().length >= 10) {
                input.setCustomValidity("");
            } else {
                input.setCustomValidity('Description must be at least 10 characters if provided');
            }
        }
    })
}

if (input.checkValidity()) {
    input.classList.remove('is-invalid');
    input.classList.add('is-valid');
} else {
    input.classList.remove('is-valid');
    input.classList.add('is-invalid');
}
});
});

```

```

        });
    }
</script>
<script>

    document.addEventListener("DOMContentLoaded", () => {

        const sb = document.getElementById("sidebar"), mc = document.querySelector(".main-content"), tb =
document.getElementById("toggleSidebar");

        tb.addEventListener("click", () => {

            const c = sb.classList.toggle("collapsed");
            mc.classList.toggle("expanded");

            const i = tb.querySelector('i');

            if (i) i.className = c ? 'bi bi-chevron-right' : 'bi bi-file';
            tb.setAttribute('aria-expanded', String(!c));
            try {localStorage.setItem('si_crud_sidebar_collapsed', String(c));} catch (e) {}

        });
        try {
            const c = localStorage.getItem('si_crud_sidebar_collapsed') === 'true';

            if (c) {sb.classList.add('collapsed'); mc.classList.add('expanded'); const i = tb.querySelector('i'); if (i) i.className = 'bi bi-chevron-right';
tb.setAttribute('aria-expanded', 'false');}

        } catch (e) {}

    });
</script>

<script>

    document.addEventListener('DOMContentLoaded', () => {

        document.querySelectorAll('.logout-link').forEach(el => {

            el.addEventListener('click', e => {

                e.preventDefault();

                Swal.fire({title:'Are you sure you want to logout?',icon:'question',showCancelButton:true,confirmButtonText:'Yes,
                logout',cancelButtonText:'Cancel'}).then(r => {if(r.isConfirmed){Swal.fire({toast:true,position:'top-end',icon:'success',title:'Successfully logged
                out',showConfirmButton:false,timer:1400,timerProgressBar:true}).then(() => {const u = new URL(el.getAttribute('href'), window.location.origin);
                u.searchParams.set('silent', '1'); window.location.href = u.toString();});}});

            });

        });
    });
</script>
</body>
</html>

```

### **Product\_Card.php**

```

<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
</head>
<style>
  body {
    font-family: monospace;
  }

  /* Sidebar style */
  .sidebar {
    width: 150px;
    min-height: 100vh;
    transition: all 0.3s ease-in-out;
    background-color: #fff;
    border-right: 1px solid #ddd;
    position: fixed;
    top: 0;
    left: 0;
    z-index: 1000;
  }

  .main-content {
    margin-left: 150px;
    transition: all 0.3s ease-in-out;
    position: relative;
  }

  /* Toggle button (outside sidebar) - fixed so it's always visible while scrolling */
  .toggle-btn {
    position: fixed;
    top: 20px;
    z-index: 2001;
    border-radius: 50%;
    width: 40px;
    height: 40px;
  }

```

```
display: flex;
align-items: center;
justify-content: center;
transition: left 0.25s ease-in-out, transform 0.25s ease-in-out;
box-shadow: 0 2px 6px rgba(0, 0, 0, 0.12);
background-color: #fff;
/* place it outside the sidebar by default (sidebar width 150px + gap 18px) */
left: 170px;
}

/* Collapsed sidebar */
.sidebar.collapsed {
margin-left: -250px;
}

.main-content.expanded {
margin-left: 0;
}

/* When sidebar is collapsed, move the toggle button to the left edge and flip icon */
.sidebar.collapsed+.main-content .toggle-btn {
left: 12px;
/* visible at the left edge */
}

.sidebar.collapsed+.main-content .toggle-btn i {
transform: rotate(180deg);
}

/* Responsive */
@media (max-width: 768px) {
.sidebar {
position: fixed;
height: 100vh;
}

.main-content {
margin-left: 0;
}
}
```

```

.nav-link.d-flex.align-items-center {
    font-size: 0.8rem;
    /* smaller text */
    font-weight: 500;
    /* optional: make it a bit bolder */
    color: #333;
    /* optional: change text color if needed */
}

.nav-link.d-flex.align-items-center:hover {
    background-color: #f8f9fa;
    color: #000;
    text-decoration: none;
}

/* make logout appear clickable */
.logout-pointer {
    cursor: pointer;
}

</style>

<body>

<body class="bg-light">

<div class="d-flex">
    <!-- Sidebar -->
    <nav id="sidebar" class="sidebar bg-white border-end">
        <div class="sidebar-header d-flex align-items-center px-3 py-3 border-bottom">
            <h5 class="mb-0"><i class="bi bi-box"></i> Product Management</h5>
        </div>
        <ul class="nav flex-column px-2 mt-3">
            <li class="nav-item mb-2">
                <a class="nav-link d-flex align-items-center" href="= base_url('/dashboard') ?">
                    <i class="bi bi-speedometer2 me-2"></i> Product Dashboard
                </a>
            </li>
            <li class="nav-item mb-2">
                <a class="nav-link d-flex align-items-center" href="= base_url('/product_card') ?">

```

```

<i class="bi bi-upc-scan me-2"></i> Products with Barcode
</a>
</li>
<li class="nav-item mb-2">
<a class="nav-link d-flex align-items-center" href="= base_url('/reports') ?&gt;"&gt;
&lt;i class="bi bi-bar-chart-line me-2"&gt;&lt;/i&gt; Reports
&lt;/a&gt;
&lt;/li&gt;
&lt;li class="nav-item mt-4"&gt;
&lt;a href="<?= base_url('/logout') ?&gt;" class="nav-link text-danger d-flex align-items-center logout-pointer logout-link"&gt;
&lt;i class="bi bi-box-arrow-right me-2"&gt;&lt;/i&gt; Logout
&lt;/a&gt;
&lt;/li&gt;

&lt;/ul&gt;
&lt;/nav&gt;
<!-- Main Content --&gt;
&lt;div class="main-content flex-grow-1 p-4"&gt;
<!-- Toggle button OUTSIDE the sidebar --&gt;
&lt;button id="toggleSidebar" class="btn btn-outline-secondary toggle-btn"&gt;
&lt;i class="bi bi-file"&gt;&lt;/i&gt;
&lt;/button&gt;
&lt;div class="d-flex"&gt;
&lt;/div&gt;
&lt;h2 class="text-center mb-3"&gt;Products Card With Barcode&lt;/h2&gt;

<!-- Product cards --&gt;
&lt;div class="container mt-4 text-center"&gt;
&lt;div class="row"&gt;
&lt;?php if (!empty($products)): ?&gt;
&lt;?php foreach ($products as $p): ?&gt;
&lt;div class="col-md-4 mb-4"&gt;
&lt;div class="card h-100 shadow-sm"&gt;
&lt;div class="card-body d-flex flex-column"&gt;
&lt;h5 class="card-title mb-2"&gt;&lt;?= esc($p['product_name']) ?&gt;&lt;/h5&gt;
&lt;p class="card-text mb-3 text-muted" style="flex:0 0 auto;"&gt;
&lt;?= esc($p['description']) ?: '&lt;span class="text-secondary"&gt;No description&lt;/span&gt;' ?&gt;
&lt;/p&gt;
&lt;p class="card-text mb-3 text-muted" style="flex:0 0 auto;"&gt;
</pre

```

```

        <?= esc($p['price']) ?: '<span class="text-secondary">No price</span>' ?>
    </p>
    <div class="mt-auto">
        <img id="pc-barcode-<?= $p['id'] ?>" alt="Barcode for product ID <?= $p['id'] ?>"/>
        <!-- View button (does not affect other behaviors) -->
        <div class="mt-3 text-center">
            <button type="button" class="btn btn-sm btn-outline-dark view-btn" data-id="<?= esc($p['id']) ?>" data-name="<?= esc($p['product_name']) ?>" data-description="<?= esc($p['description']) ?>" data-created="<?= date('M d, Y', strtotime($p['created_at'])) ?>" data-price="<?= esc($p['price']) ?>"><i class="bi bi-search"></i> View</button>
        </div>
        <div class="small text-muted mt-2">Created: <?= date('M d, Y', strtotime($p['created_at'])) ?></div>
    </div>
    </div>
</div>
<?php endforeach; ?>
<?php else: ?>
    <div class="col-12 text-center py-4">
        <p class="text-muted">No products found.</p>
    </div>
<?php endif; ?>
</div>
</div>

<script>
    document.addEventListener("DOMContentLoaded", function() {
        const sidebar = document.getElementById("sidebar");
        const mainContent = document.querySelector(".main-content");
        const toggleBtn = document.getElementById("toggleSidebar");

        toggleBtn.addEventListener("click", () => {
            const isCollapsed = sidebar.classList.toggle("collapsed");
            mainContent.classList.toggle("expanded");
            // swap icon
            const icon = toggleBtn.querySelector('i');
            if (icon) icon.className = isCollapsed ? 'bi bi-chevron-right' : 'bi bi-file';
            // aria
            toggleBtn.setAttribute('aria-expanded', String(!isCollapsed));
            // persist
            try {
                localStorage.setItem('si_crud_sidebar_collapsed', String(isCollapsed));
            }
        });
    });
</script>

```

```

        } catch (e) {}

    });

    // Restore state on load
    try {

        const collapsed = localStorage.getItem('si_crud_sidebar_collapsed') === 'true';

        if (collapsed) {

            sidebar.classList.add('collapsed');

            mainContent.classList.add('expanded');

            const icon = toggleBtn.querySelector('i');

            if (icon) icon.className = 'bi bi-chevron-right';

            toggleBtn.setAttribute('aria-expanded', 'false');

        }

    } catch (e) {}

});

</script>
</script>

<!-- JsBarcode CDN -->

<script src="https://cdn.jsdelivr.net/npm/jsbarcode@3.11.5/dist/JsBarcode.all.min.js"></script>

<script>

    document.addEventListener('DOMContentLoaded', function() {

        <?php if (!empty($products)): ?>

        <?php foreach ($products as $p): ?>

        try {

            // card barcode

            JsBarcode('#pc-barcode-<?= $p['id'] ?>', '<?= esc($p['id']) ?>', {

                format: 'CODE128',
                width: 2,
                height: 40,
                displayValue: true
            });

            // modal barcode (slightly taller for visibility)

            try {

                JsBarcode('#pc-modal-barcode-<?= $p['id'] ?>', '<?= esc($p['id']) ?>', {

                    format: 'CODE128',
                    width: 2,
                    height: 60,
                    displayValue: true
                });

            } catch (e) {}

        } catch (e) {}

    });


```

```

        } catch (e) {}

        <?php endforeach; ?>

        <?php endif; ?>

    });

</script>

<!-- Use Bootstrap modal for smooth open/close (include bundle for modal) --&gt;
&lt;script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"&gt;&lt;/script&gt;

<!-- Shared Bootstrap modal (single instance) --&gt;
&lt;div class="modal fade" id="pcSharedModal" tabindex="-1" aria-labelledby="pcSharedModalLabel" aria-hidden="true"&gt;
    &lt;div class="modal-dialog modal-dialog-centered"&gt;
        &lt;div class="modal-content"&gt;
            &lt;div class="modal-header"&gt;
                &lt;h5 class="modal-title" id="pcSharedModalLabel"&gt;Product details&lt;/h5&gt;
                &lt;button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"&gt;&lt;/button&gt;
            &lt;/div&gt;
            &lt;div class="modal-body text-center"&gt;
                &lt;p&gt;&lt;strong&gt;Name:&lt;/strong&gt; &lt;span id="pc-modal-name"&gt;&lt;/span&gt;&lt;/p&gt;
                &lt;p&gt;&lt;strong&gt;Description:&lt;/strong&gt; &lt;span id="pc-modal-description"&gt;&lt;/span&gt;&lt;/p&gt;
                &lt;p&gt;&lt;strong&gt;price:&lt;/strong&gt; &lt;span id="pc-modal-price"&gt;&lt;/span&gt;&lt;/p&gt;
                &lt;p&gt;&lt;strong&gt;Created:&lt;/strong&gt; &lt;span id="pc-modal-created"&gt;&lt;/span&gt;&lt;/p&gt;
                &lt;p&gt;&lt;strong&gt;ID:&lt;/strong&gt; &lt;span id="pc-modal-id"&gt;&lt;/span&gt;&lt;/p&gt;
                &lt;div class="text-center mt-3"&gt;
                    &lt;img id="pc-shared-modal-barcode" alt="Barcode" aria-hidden="true"/>
                </div>
            </div>
        </div>
    </div>
</div>

<script>
    // Fill and show Bootstrap modal on view button click, generate barcode on show
    (function() {
        var pcSharedModalEl = document.getElementById('pcSharedModal');

        var bsModal = new bootstrap.Modal(pcSharedModalEl, {
            backdrop: true
        });

        document.addEventListener('click', function(e) {

```

```

var t = e.target.closest('.view-btn');

if (!t) return;

var id = t.getAttribute('data-id');

var name = t.getAttribute('data-name') || "";

var desc = t.getAttribute('data-description') || "";

var created = t.getAttribute('data-created') || "";

var price = t.getAttribute('data-price') || "";

document.getElementById('pc-modal-name').textContent = name;

document.getElementById('pc-modal-description').textContent = desc || 'No description';

document.getElementById('pc-modal-created').textContent = created;

document.getElementById('pc-modal-id').textContent = id;

document.getElementById('pc-modal-price').textContent = price || 'No price';

// generate barcode into shared modal svg (try/catch so it won't break)

try {

  JsBarcode('#pc-shared-modal-barcode', String(id), {

    format: 'CODE128',
    width: 2,
    height: 60,
    displayValue: true
  });

} catch (e) {}

bsModal.show();

}, false);

})();

</script>

<script>

// Logout confirmation using SweetAlert2

document.addEventListener('DOMContentLoaded', function() {

  document.querySelectorAll('.logout-link').forEach(function(el) {

    el.addEventListener('click', function(e) {

      e.preventDefault();

      const href = this.getAttribute('href');

      Swal.fire({

        title: 'Are you sure you want to logout?',
        icon: 'question',
        showCancelButton: true,

```

```

        confirmButtonText: 'Yes, logout',
        cancelButtonText: 'Cancel'

    }).then((result) => {
        if (result.isConfirmed) {
            // Show a short success toast then redirect to logout with ?silent=1
            Swal.fire({
                toast: true,
                position: 'top-end',
                icon: 'success',
                title: 'Successfully logged out',
                showConfirmButton: false,
                timer: 1400,
                timerProgressBar: true,
            }).then(() => {
                // append silent param to avoid server flash duplication
                const url = new URL(href, window.location.origin);
                url.searchParams.set('silent', '1');
                window.location.href = url.toString();
            });
        }
    });
});

});

});

});

});

};


```

</script>

</body>

</html>

### **Reports.php**

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Reports - Product Management</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
    <script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.0/dist/chart.umd.min.js"></script>

```

```
</head>

<style>

body {
    font-family: monospace;
}

/* Sidebar style */
.sidebar {
    width: 150px;
    min-height: 100vh;
    transition: all 0.3s ease-in-out;
    background-color: #fff;
    border-right: 1px solid #ddd;
    position: fixed;
    top: 0;
    left: 0;
    z-index: 1000;
}

.main-content {
    margin-left: 150px;
    transition: all 0.3s ease-in-out;
    position: relative;
}

.toggle-btn {
    position: fixed;
    top: 20px;
    z-index: 2001;
    border-radius: 50%;
    width: 40px;
    height: 40px;
    display: flex;
    align-items: center;
    justify-content: center;
    transition: left 0.25s ease-in-out, transform 0.25s ease-in-out;
    box-shadow: 0 2px 6px rgba(0, 0, 0, 0.12);
    background-color: #fff;
    left: 170px;
}
```

```
.sidebar.collapsed {  
    margin-left: -250px;  
}  
  
.main-content.expanded {  
    margin-left: 0;  
}  
  
.sidebar.collapsed+.main-content .toggle-btn {  
    left: 12px;  
}  
  
.sidebar.collapsed+.main-content .toggle-btn i {  
    transform: rotate(180deg);  
}  
  
@media (max-width: 768px) {  
    .sidebar {  
        position: fixed;  
        height: 100vh;  
    }  
  
    .main-content {  
        margin-left: 0;  
    }  
}  
  
.nav-link.d-flex.align-items-center {  
    font-size: 0.8rem;  
    font-weight: 500;  
    color: #333;  
}  
  
.nav-link.d-flex.align-items-center:hover {  
    background-color: #f8f9fa;  
    color: #000;  
    text-decoration: none;  
}
```

```

.logout-pointer {
  cursor: pointer;
}

</style>

<body class="bg-light">
<div class="d-flex">

<div class="d-flex">
  <!-- Sidebar -->
  <nav id="sidebar" class="sidebar bg-white border-end">
    <div class="sidebar-header d-flex align-items-center px-3 py-3 border-bottom">
      <h5 class="mb-0"><i class="bi bi-box"></i> Product Management</h5>
    </div>
    <ul class="nav flex-column px-2 mt-3">
      <li class="nav-item mb-2">
        <a class="nav-link d-flex align-items-center" href="= base_url('/dashboard') ?">
          <i class="bi bi-speedometer2 me-2"></i> Product Dashboard
        </a>
      </li>
      <li class="nav-item mb-2">
        <a class="nav-link d-flex align-items-center" href="= base_url('/product_card') ?">
          <i class="bi bi-upc-scan me-2"></i> Products with Barcode
        </a>
      </li>
      <li class="nav-item mb-2">
        <a class="nav-link d-flex align-items-center" href="= base_url('/reports') ?">
          <i class="bi bi-bar-chart-line me-2"></i> Reports
        </a>
      </li>
      <li class="nav-item mt-4">
        <a href="= base_url('/logout') ?" class="nav-link text-danger d-flex align-items-center logout-pointer logout-link">
          <i class="bi bi-box-arrow-right me-2"></i> Logout
        </a>
      </li>
    </ul>
  </nav>
  <!-- Main Content -->
</div>

```

```

<div class="main-content flex-grow-1 p-4">
    <button id="toggleSidebar" class="btn btn-outline-secondary toggle-btn"><i class="bi bi-file"></i></button>

    <div class="container-fluid">
        <h2 class="mb-3 text-center">Reports & Analytics</h2>

        <?php
            $totalProducts = isset($totalCount) ? (int)$totalCount : (isset($products) ? count($products) : 0);
            $today = date('Y-m-d');
            $createdToday = 0;
            $avgPrice = 0.0;

            $productsList = isset($products) ? $products : [];

            $priceSum = 0.0;
            $priceCount = 0;
            if (!empty($productsList)) {
                foreach ($productsList as $p) {
                    if (isset($p['price']) && $p['price'] !== "") {
                        $val = floatval($p['price']);
                        if (is_numeric($val)) {
                            $priceSum += $val;
                            $priceCount++;
                        }
                    }
                    if (!empty($p['created_at'])) {
                        if (date('Y-m-d', strtotime($p['created_at'])) === $today) {
                            $createdToday++;
                        }
                    }
                }
            }

            if ($priceCount > 0) {
                $avgPrice = $priceSum / $priceCount;
            }

            $monthlyLabels = [];
            $monthlyData = [];
            for ($i = 5; $i >= 0; $i--) {

```

```

$m = date('Y-m', strtotime("-{$i} month"));

$label = date('M Y', strtotime($m . '-01'));

$monthlyLabels[] = $label;

$monthlyData[$m] = 0;

}

if (!empty($productsList)) {

foreach ($productsList as $pr) {

if (!empty($pr['created_at'])) {

$key = date('Y-m', strtotime($pr['created_at']));

if (array_key_exists($key, $monthlyData)) $monthlyData[$key]++;

}

}

}

$monthlyCounts = array_values($monthlyData);

$topProducts = $productsList;

usort($topProducts, function ($a, $b) {

return floatval($b['price'] ?? 0) <=> floatval($a['price'] ?? 0);

});

$topProducts = array_slice($topProducts, 0, 10);

?>

<!-- Summary cards -->

<div class="row g-3 mb-4">

<div class="col-12 col-md-4">

<div class="card shadow-sm">

<div class="card-body">

<h6 class="card-title mb-1">Total Products</h6>

<h3 class="mb-0"><?php echo $totalProducts ?></h3>

<small class="text-muted">All time</small>

</div>

</div>

<div class="col-12 col-md-4">

<div class="card shadow-sm">

<div class="card-body">

<h6 class="card-title mb-1">Created Today</h6>

<h3 class="mb-0"><?php echo $createdToday ?></h3>

<small class="text-muted"><?php echo $today ?></small>

</div>

```

```

</div>
</div>

<div class="col-12 col-md-4">
  <div class="card shadow-sm">
    <div class="card-body">
      <h6 class="card-title mb-1">Average Price</h6>
      <h3 class="mb-0">₱<?php echo number_format($avgPrice, 2) ?></h3>
      <small class="text-muted">Across listed products</small>
    </div>
  </div>
</div>

<div>
  <!-- Charts -->
  <div class="row g-3 mb-4">
    <div class="col-12 col-md-12">
      <div class="card shadow-sm h-100 chart-card">
        <div class="card-body d-flex flex-column">
          <h6 class="card-title">Products Added (Last 6 months)</h6>
          <div class="flex-grow-1">
            <canvas id="lineMonthly" style="width:100%;height:100%;"></canvas>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="col-12 col-md-12">
    <div class="card shadow-sm h-100 chart-card">
      <div class="card-body d-flex flex-column">
        <h6 class="card-title">Top Priced Products</h6>
        <div class="d-flex justify-content-between align-items-center mb-2">
          <div></div>
          <div>
            <button id="downloadTopCsv" class="btn btn-sm btn-outline-primary"><i class="bi bi-download"></i> Download CSV</button>
          </div>
        </div>
      </div>
    </div>
    <div class="table-responsive">
      <table class="table table-sm table-hover mb-0">
        <thead>
          <tr>
            <th>Name</th>

```

```

<th class="text-end">Price</th>
</tr>
</thead>
<tbody>
<?php if (!empty($topProducts)): ?>
<?php foreach ($topProducts as $tp): ?>
<tr>
<td><?= esc($tp['product_name'] ?? $tp['name'] ?? '-') ?></td>
<td class="text-end">₱<?= number_format(floatval($tp['price'] ?? 0), 2) ?></td>
</tr>
<?php endforeach; ?>
<?php else: ?>
<tr>
<td colspan="2" class="text-center">No products available</td>
</tr>
<?php endif; ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div> <!-- container-fluid -->
</div> <!-- main-content -->
</div> <!-- d-flex -->

<script>
const monthlyLabels = <?= json_encode($monthlyLabels) ?>;
const monthlyCounts = <?= json_encode($monthlyCounts) ?>;

function generateColors(n) {
  const palette = ['#4e73df', '#1cc88a', '#36b9cc', '#f6c23e', '#e74a3b', '#858796', '#6f42c1', '#fd7e14', '#20c997', '#0dcraf0'];
  const out = [];
  for (let i = 0; i < n; i++) out.push(palette[i % palette.length]);
  return out;
}

const ctxLine = document.getElementById('lineMonthly').getContext('2d');

```

```

new Chart(ctxLine, {
    type: 'line',
    data: {
        labels: monthlyLabels,
        datasets: [
            {
                label: 'Products added',
                data: monthlyCounts,
                borderColor: '#4e73df',
                backgroundColor: 'rgba(78,115,223,0.12)',
                tension: 0.3,
                fill: true,
                pointRadius: 4
            }
        ],
        options: {
            responsive: true,
            maintainAspectRatio: false,
            plugins: {
                legend: {
                    display: false
                }
            },
            scales: {
                x: {
                    ticks: {
                        maxRotation: 0,
                        autoSkip: true
                    }
                },
                y: {
                    beginAtZero: true,
                    precision: 0
                }
            }
        }
    });
</script>

<script>
const topProducts = <?= json_encode($topProducts) ?> || [];

```

```

function convertToCSV(items) {
    if (!items || !items.length) return "";

    const header = ['name', 'price'];

    const lines = [header.join(',')];

    items.forEach(it => {
        const name = (it.product_name ?? it.name ?? "").toString().replace(/\"/g, "'");
        const price = parseFloat(it.price ?? 0).toFixed(2);

        const esc = (v) => (/[^,\n]/.test(v) ? "" + v.replace(/\"/g, "") : v);

        lines.push([esc(name), esc(price)].join(','));
    });

    return lines.join("\r\n");
}

function downloadCSV(filename, csvContent) {
    const blob = new Blob([csvContent], {
        type: 'text/csv;charset=utf-8';
    });

    if (navigator.msSaveBlob) {
        navigator.msSaveBlob(blob, filename);
    } else {
        const link = document.createElement('a');

        if (link.download !== undefined) {
            const url = URL.createObjectURL(blob);

            link.setAttribute('href', url);
            link.setAttribute('download', filename);
            link.style.visibility = 'hidden';

            document.body.appendChild(link);
            link.click();
            document.body.removeChild(link);
            URL.revokeObjectURL(url);
        }
    }
}

document.addEventListener('DOMContentLoaded', function() {
    const btn = document.getElementById('downloadTopCsv');

    if (!btn) return;

    if (!topProducts || topProducts.length === 0) {
        btn.disabled = true;
    }
});

```

```

btn.title = 'No top products to export';

return;

}

btn.addEventListener('click', function() {

const csv = convertToCSV(topProducts);

const now = new Date();

const timestamp = now.toISOString().slice(0, 19).replace(/:/g, '-');

downloadCSV('top-prices-' + timestamp + '.csv', csv);

});

});

</script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

<script>

document.addEventListener("DOMContentLoaded", function() {

const sidebar = document.getElementById("sidebar");

const mainContent = document.querySelector(".main-content");

const toggleBtn = document.getElementById("toggleSidebar");

toggleBtn.addEventListener("click", () => {

const isCollapsed = sidebar.classList.toggle("collapsed");

mainContent.classList.toggle("expanded");

const icon = toggleBtn.querySelector('i');

if (icon) icon.className = isCollapsed ? 'bi bi-chevron-right' : 'bi bi-file';

try {

localStorage.setItem('si_crud_sidebar_collapsed', String(isCollapsed));

} catch (e) {}

});

try {

const collapsed = localStorage.getItem('si_crud_sidebar_collapsed') === 'true';

if (collapsed) {

sidebar.classList.add('collapsed');

mainContent.classList.add('expanded');

const icon = toggleBtn.querySelector('i');

if (icon) icon.className = 'bi bi-chevron-right';

}

} catch (e) {}

});

document.addEventListener('DOMContentLoaded', function() {

```

```

document.querySelectorAll('.logout-link').forEach(function(el) {
    el.addEventListener('click', function(e) {
        e.preventDefault();
        const href = this.getAttribute('href');
        Swal.fire({
            title: 'Are you sure you want to logout?',
            icon: 'question',
            showCancelButton: true,
            confirmButtonText: 'Yes, logout',
            cancelButtonText: 'Cancel'
        }).then((result) => {
            if (result.isConfirmed) {
                Swal.fire({
                    toast: true,
                    position: 'top-end',
                    icon: 'success',
                    title: 'Successfully logged out',
                    showConfirmButton: false,
                    timer: 1400,
                    timerProgressBar: true
                }).then(() => {
                    try {
                        const url = new URL(href, window.location.origin);
                        url.searchParams.set('silent', '1');
                        window.location.href = url.toString();
                    } catch (err) {
                        window.location.href = href + (href.includes('?') ? '&' : '?') + 'silent=1';
                    }
                });
            }
        });
    });
});

```

</script>

</body>

</html>

### Reports.php

```

<!DOCTYPE html>
<html lang="en">

```

```
<head>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Reports - Product Management</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
    <script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.0/dist/chart.umd.min.js"></script>

</head>

<style>

body {
    font-family: monospace;
}

/* Sidebar style */
.sidebar {
    width: 150px;
    min-height: 100vh;
    transition: all 0.3s ease-in-out;
    background-color: #fff;
    border-right: 1px solid #ddd;
    position: fixed;
    top: 0;
    left: 0;
    z-index: 1000;
}

.main-content {
    margin-left: 150px;
    transition: all 0.3s ease-in-out;
    position: relative;
}

.toggle-btn {
    position: fixed;
    top: 20px;
    z-index: 2001;
    border-radius: 50%;
    width: 40px;
}
```

```
height: 40px;  
display: flex;  
align-items: center;  
justify-content: center;  
transition: left 0.25s ease-in-out, transform 0.25s ease-in-out;  
box-shadow: 0 2px 6px rgba(0, 0, 0, 0.12);  
background-color: #fff;  
left: 170px;  
}
```

```
.sidebar.collapsed {  
margin-left: -250px;  
}
```

```
.main-content.expanded {  
margin-left: 0;  
}
```

```
.sidebar.collapsed+.main-content .toggle-btn {  
left: 12px;  
}
```

```
.sidebar.collapsed+.main-content .toggle-btn i {  
transform: rotate(180deg);  
}
```

```
@media (max-width: 768px) {  
.sidebar {  
position: fixed;  
height: 100vh;  
}
```

```
.main-content {  
margin-left: 0;  
}  
}
```

```
.nav-link.d-flex.align-items-center {  
font-size: 0.8rem;  
font-weight: 500;
```

```

color: #333;
}

.nav-link.d-flex.align-items-center:hover {
background-color: #f8f9fa;
color: #000;
text-decoration: none;
}

.logout-pointer {
cursor: pointer;
}

</style>

<body class="bg-light">
<div class="d-flex">

<div class="d-flex">
<!-- Sidebar -->
<nav id="sidebar" class="sidebar bg-white border-end">
<div class="sidebar-header d-flex align-items-center px-3 py-3 border-bottom">
<h5 class="mb-0"><i class="bi bi-box"></i> Product Management</h5>
</div>
<ul class="nav flex-column px-2 mt-3">
<li class="nav-item mb-2">
<a class="nav-link d-flex align-items-center" href="<?= base_url('/dashboard') ?>">
<i class="bi bi-speedometer2 me-2"></i> Product Dashboard
</a>
</li>
<li class="nav-item mb-2">
<a class="nav-link d-flex align-items-center" href="<?= base_url('/product_card') ?>">
<i class="bi bi-upc-scan me-2"></i> Products with Barcode
</a>
</li>
<li class="nav-item mb-2">
<a class="nav-link d-flex align-items-center" href="<?= base_url('/reports') ?>">
<i class="bi bi-bar-chart-line me-2"></i> Reports
</a>
</li>
<li class="nav-item mt-4">

```

```

<a href="= base_url('/logout') ?" class="nav-link text-danger d-flex align-items-center logout-pointer logout-link">
    <i class="bi bi-box-arrow-right me-2"></i> Logout
</a>
</li>

</ul>
</nav>

<!-- Main Content -->
<div class="main-content flex-grow-1 p-4">
    <button id="toggleSidebar" class="btn btn-outline-secondary toggle-btn"><i class="bi bi-file"></i></button>

    <div class="container-fluid">
        <h2 class="mb-3 text-center">Reports & Analytics</h2>

        <?php
        $totalProducts = isset($totalCount) ? (int)$totalCount : (isset($products) ? count($products) : 0);
        $today = date('Y-m-d');
        $createdToday = 0;
        $avgPrice = 0.0;

        $productsList = isset($products) ? $products : [];

        $priceSum = 0.0;
        $priceCount = 0;
        if (!empty($productsList)) {
            foreach ($productsList as $p) {
                if (isset($p['price']) && $p['price'] !== "") {
                    $val = floatval($p['price']);
                    if (is_numeric($val)) {
                        $priceSum += $val;
                        $priceCount++;
                    }
                }
            }
            if (!empty($p['created_at'])) {
                if (date('Y-m-d', strtotime($p['created_at'])) === $today) {
                    $createdToday++;
                }
            }
        }
    
```

```

}

if ($priceCount > 0) {
    $avgPrice = $priceSum / $priceCount;
}

$monthlyLabels = [];
$monthlyData = [];
for ($i = 5; $i >= 0; $i--) {
    $m = date('Y-m', strtotime("-{$i} month"));
    $label = date('M Y', strtotime($m . '-01'));
    $monthlyLabels[] = $label;
    $monthlyData[$m] = 0;
}
if (!empty($productsList)) {
    foreach ($productsList as $pr) {
        if (!empty($pr['created_at'])) {
            $key = date('Y-m', strtotime($pr['created_at']));
            if (array_key_exists($key, $monthlyData)) $monthlyData[$key]++;
        }
    }
}
$monthlyCounts = array_values($monthlyData);

$topProducts = $productsList;
usort($topProducts, function ($a, $b) {
    return floatval($b['price'] ?? 0) <=> floatval($a['price'] ?? 0);
});
$topProducts = array_slice($topProducts, 0, 10);
?>

<!-- Summary cards -->
<div class="row g-3 mb-4">
    <div class="col-12 col-md-4">
        <div class="card shadow-sm">
            <div class="card-body">
                <h6 class="card-title mb-1">Total Products</h6>
                <h3 class="mb-0"><?php echo $totalProducts ?></h3>
                <small class="text-muted">All time</small>
            </div>
        </div>
    </div>
</div>
```

```

        </div>
    </div>

<div class="col-12 col-md-4">
    <div class="card shadow-sm">
        <div class="card-body">
            <h6 class="card-title mb-1">Created Today</h6>
            <h3 class="mb-0"><?php echo $createdToday ?></h3>
            <small class="text-muted"><?php echo $today ?></small>
        </div>
    </div>
</div>

<div class="col-12 col-md-4">
    <div class="card shadow-sm">
        <div class="card-body">
            <h6 class="card-title mb-1">Average Price</h6>
            <h3 class="mb-0"><?php echo number_format($avgPrice, 2) ?></h3>
            <small class="text-muted">Across listed products</small>
        </div>
    </div>
</div>

<!-- Charts -->
<div class="row g-3 mb-4">
    <div class="col-12 col-md-12">
        <div class="card shadow-sm h-100 chart-card">
            <div class="card-body d-flex flex-column">
                <h6 class="card-title">Products Added (Last 6 months)</h6>
                <div class="flex-grow-1">
                    <canvas id="lineMonthly" style="width:100%;height:100%;"></canvas>
                </div>
            </div>
        </div>
    </div>
    <div class="col-12 col-md-12">
        <div class="card shadow-sm h-100 chart-card">
            <div class="card-body d-flex flex-column">
                <h6 class="card-title">Top Priced Products</h6>
                <div class="d-flex justify-content-between align-items-center mb-2">
                    <div></div>

```



```

function generateColors(n) {
  const palette = ['#4e73df', '#1cc88a', '#36b9cc', '#f6c23e', '#e74a3b', '#858796', '#6f42c1', '#fd7e14', '#20c997', '#0dcacf0'];
  const out = [];
  for (let i = 0; i < n; i++) out.push(palette[i % palette.length]);
  return out;
}

const ctxLine = document.getElementById('lineMonthly').getContext('2d');

new Chart(ctxLine, {
  type: 'line',
  data: {
    labels: monthlyLabels,
    datasets: [
      {
        label: 'Products added',
        data: monthlyCounts,
        borderColor: '#4e73df',
        backgroundColor: 'rgba(78,115,223,0.12)',
        tension: 0.3,
        fill: true,
        pointRadius: 4
      }
    ],
    options: {
      responsive: true,
      maintainAspectRatio: false,
      plugins: {
        legend: {
          display: false
        }
      },
      scales: {
        x: {
          ticks: {
            maxRotation: 0,
            autoSkip: true
          }
        },
        y: {
          beginAtZero: true,

```

```

precision: 0
}
}
}
});
</script>

<script>
const topProducts = <?= json_encode($topProducts) ?> || [];

function convertToCSV(items) {
    if (!items || !items.length) return "";
    const header = ['name', 'price'];
    const lines = [header.join(',')];
    items.forEach(it => {
        const name = (it.product_name ?? it.name ?? "").toString().replace(/\"/g, "''");
        const price = parseFloat(it.price ?? 0).toFixed(2);
        const esc = (v) => (/[^,\n]/.test(v) ? "" + v.replace(/\"/g, "''") + "" : v);
        lines.push([esc(name), esc(price)].join(','));
    });
    return lines.join("\r\n");
}

function downloadCSV(filename, csvContent) {
    const blob = new Blob([csvContent], {
        type: 'text/csv;charset=utf-8';
    });
    if (navigator.msSaveBlob) {
        navigator.msSaveBlob(blob, filename);
    } else {
        const link = document.createElement('a');
        if (link.download !== undefined) {
            const url = URL.createObjectURL(blob);
            link.setAttribute('href', url);
            link.setAttribute('download', filename);
            link.style.visibility = 'hidden';
            document.body.appendChild(link);
            link.click();
            document.body.removeChild(link);
            URL.revokeObjectURL(url);
        }
    }
}

```

```

        }
    }
}

document.addEventListener('DOMContentLoaded', function() {
    const btn = document.getElementById('downloadTopCsv');
    if (!btn) return;
    if (!topProducts || topProducts.length === 0) {
        btn.disabled = true;
        btn.title = 'No top products to export';
        return;
    }
    btn.addEventListener('click', function() {
        const csv = convertToCSV(topProducts);
        const now = new Date();
        const timestamp = now.toISOString().slice(0, 19).replace(/:/g, '-');
        downloadCSV('top-prices-' + timestamp + '.csv', csv);
    });
});

```

</script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

```

<script>

    document.addEventListener("DOMContentLoaded", function() {
        const sidebar = document.getElementById("sidebar");
        const mainContent = document.querySelector(".main-content");
        const toggleBtn = document.getElementById("toggleSidebar");
        toggleBtn.addEventListener("click", () => {
            const isCollapsed = sidebar.classList.toggle("collapsed");
            mainContent.classList.toggle("expanded");
            const icon = toggleBtn.querySelector('i');
            if (icon) icon.className = isCollapsed ? 'bi bi-chevron-right' : 'bi bi-file';
            try {
                localStorage.setItem('si_crud_sidebar_collapsed', String(isCollapsed));
            } catch (e) {}
        });
        try {
            const collapsed = localStorage.getItem('si_crud_sidebar_collapsed') === 'true';
            if (collapsed) {

```

```

        sidebar.classList.add('collapsed');

        mainContent.classList.add('expanded');

        const icon = toggleBtn.querySelector('i');

        if (icon) icon.className = 'bi bi-chevron-right';

    }

} catch (e) {}

});

document.addEventListener('DOMContentLoaded', function() {

    document.querySelectorAll('.logout-link').forEach(function(el) {

        el.addEventListener('click', function(e) {

            e.preventDefault();

            const href = this.getAttribute('href');

            Swal.fire({

                title: 'Are you sure you want to logout?',
                icon: 'question',
                showCancelButton: true,
                confirmButtonText: 'Yes, logout',
                cancelButtonText: 'Cancel'

            }).then((result) => {

                if (result.isConfirmed) {

                    Swal.fire({

                        toast: true,
                        position: 'top-end',
                        icon: 'success',
                        title: 'Successfully logged out',
                        showConfirmButton: false,
                        timer: 1400,
                        timerProgressBar: true

                    }).then(() => {

                        try {

                            const url = new URL(href, window.location.origin);

                            url.searchParams.set('silent', '1');

                            window.location.href = url.toString();

                        } catch (err) {

                            window.location.href = href + (href.includes('?') ? '&' : '?') + 'silent=1';

                        }

                    });

                }

            });

        });

    });

});
```

```
    });
    });
});
</script>
</body>
</html>
```

**Pager/ bootstrap\_pagination.php**

```
<style>
```

```
.pagination .page-link {  
    color: #212529;  
    border: 1px solid #212529;  
    background-color: transparent;  
}
```

```
.pagination .page-link:hover {
```

```
    color: #fff;  
    background-color: #212529;  
    border-color: #212529;  
}
```

```
.pagination .page-link:focus {
```

```
    color: #fff;  
    background-color: #000;  
    border-color: #000;  
    box-shadow: 0 0 0 0.2rem rgba(0, 0, 0, 0.25);  
}
```

```
.pagination .page-item.active .page-link {
```

```
    color: #fff;  
    background-color: #212529;  
    border-color: #212529;  
}
```

```
</style>
```

```
<?php $pager->setSurroundCount(2) ?>
```

```
<nav aria-label="<?= lang('Pager.pageNavigation') ?>">
```

```
<ul class="pagination justify-content-center">
```

```
<?php if ($pager->hasPrevious()) : ?>
```

```
    <li class="page-item">
```

```
        <a class="page-link" href="<?= $pager->getFirst() ?>" aria-label="<?= lang('Pager.first') ?>">
```

```
            <span aria-hidden="true"><?= lang('Pager.first') ?></span>
```

```
        </a>
```

```
    </li>
```

```
    <li class="page-item">
```

```

<a class="page-link" href="= $pager-&gt;getPrevious() ?&gt;" aria-label="<?= lang('Pager.previous') ?&gt;""
    <span aria-hidden="true">&laquo;</span>
</a>
</li>
<?php endif ?>

<?php foreach ($pager->links() as $link) : ?>
<li class="page-item <?= $link['active'] ? 'active' : '' ?>">
    <a class="page-link" href="= $link['uri'] ?&gt;"&gt;
        &lt;?= $link['title'] ?&gt;
    &lt;/a&gt;
&lt;/li&gt;
&lt;?php endforeach ?&gt;

&lt;?php if ($pager-&gt;hasNext()) : ?&gt;
&lt;li class="page-item"&gt;
    &lt;a class="page-link" href="<?= $pager-&gt;getNext() ?&gt;" aria-label="<?= lang('Pager.next') ?&gt;""
        <span aria-hidden="true">&raquo;</span>
    </a>
</li>
<li class="page-item">
    <a class="page-link" href="= $pager-&gt;getLast() ?&gt;" aria-label="<?= lang('Pager.last') ?&gt;""
        <span aria-hidden="true"><?= lang('Pager.last') ?></span>
    </a>
</li>
<?php endif ?>
</ul>
</nav>

```

#### Email/ reset\_password.php

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>

```

```
<p>Hello,</p>

<p>You requested a password reset. Click the link below to set a new password. The link will expire in <?= esc($expires) ?> minutes.</p>

<p><a href="<?= esc($link) ?>"><?= esc($link) ?></a></p>

<p>If you did not request this, ignore this email.</p>

</body>

</html>
```

```
Auth/ forgot_password.php

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Forgot Password</title>

<!-- Bootstrap 5 CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-sRl4kxLFvY47J16cr9ZwB07vP4J8+LH7qKQnuqkulAvNWLzeN8tE5YBujZqJLB" crossorigin="anonymous">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet" integrity="sha384-tViUuhnYAV00FLlhhi3v/dWt3Jxw4gZQcNoSCxCIFNJVCx7/D55/wXsrNIRANwdD" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

<style>

.form-control:focus {

    box-shadow: none;

    outline: none;

    border-color: #aaa;

    /* optional: custom border color */

}
```

}

```
</style>

</head>

<body class="bg-light">

<div class="d-flex align-items-center justify-content-center" style="height:100vh;">

<div class="card shadow-sm" style="min-width: 350px;">

<div class="card-body">
```

```

<h4 class="card-title mb-3 text-center">Forgot Password</h4>

<?php if (session()->getFlashdata('success')): ?>
<script>
Swal.fire({
  icon: 'success',
  title: <?= json_encode(session()->getFlashdata('success')) ?>,
  timer: 2000,
  showConfirmButton: false
});
</script>
<?php endif; ?>

<?php if (session()->getFlashdata('error')): ?>
<script>
Swal.fire({
  icon: 'error',
  title: <?= json_encode(session()->getFlashdata('error')) ?>
});
</script>
<?php endif; ?>

<form action=<?= base_url('forgot-password') ?>" method="post">
<div class="mb-3">
  <label for="email" class="form-label">Email address</label>
  <input type="email" class="form-control" id="email" name="email" required>
</div>
<div class="d-grid">
  <button type="submit" class="btn btn-outline-dark">Send Reset Link</button>
</div>
</form>

<div class="mt-3 text-center">
  <a class="text-decoration-none text-dark" href=<?= base_url('/') ?>>Back to login</a>
</div>
</div>
</div>

<!-- Bootstrap 5 JS bundle -->
```

```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjdO4Dr2bkBIFxQpeoA6V2b6a6Q9p5Q5p5Q5p5" crossorigin="anonymous"></script>
</body>

</html>

Auth/ reset_password.php

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Reset Password</title>
<!-- Bootstrap 5 CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-SRI4kxILFvY47J16cr9ZwB07vP4J8+LH7qKQnuqkulAvNWLzeN8tE5YBujZqJLB" crossorigin="anonymous">
<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" integrity="sha384-tViUubbYAV00FLIhhi3v/dWt3Jxw4gZQcNoSCxCIFNJVCx7/D55/wXsrNIRANwdD" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
<style>
.form-control:focus {
  box-shadow: none;
  outline: none;
  border-color: #aaa;
  /* optional: custom border color */
}

</style>
</head>

<body class="bg-light">

<div class="container py-5">
<div class="row justify-content-center">
<div class="col-md-6">
<div class="card shadow-sm">
<div class="card-body">
<h4 class="card-title mb-3">Reset Password</h4>

<?php if (session()->getFlashdata('success')): ?>

```

```

<script>
Swal.fire({
icon: 'success',
title: <?= json_encode(session()->getFlashdata('success')) ?>,
timer: 2000,
showConfirmButton: false
});
</script>

<?php endif; ?>

<?php if (session()->getFlashdata('error')): ?>
<script>
Swal.fire({
icon: 'error',
title: <?= json_encode(session()->getFlashdata('error')) ?>
});
</script>
<?php endif; ?>

<form action=<?= base_url('reset-password') ?>" method="post">
<input type="hidden" name="token" value=<?= esc($token) ?>">
<div class="mb-3">
<label for="password" class="form-label">New password</label>
<input type="password" class="form-control" id="password" name="password" required minlength="6">
</div>
<div class="mb-3">
<label for="password_confirm" class="form-label">Confirm password</label>
<input type="password" class="form-control" id="password_confirm" name="password_confirm" required>
</div>
<div class="d-grid">
<button type="submit" class="btn btn-outline-dark">Set Password</button>
</div>
</form>

<div class="mt-3 text-center">
<a class="text-decoration-underline text-dark" href=<?= base_url('/') ?>">Back to login</a>
</div>
</div>
</div>
</div>

```

```
</div>

<!-- Bootstrap 5 JS bundle -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjdO4Dr2bkBIFxQpeoA6V2b6a6Q9p5Q5p5Q5p5" crossorigin="anonymous"></script>

</body>

</html>
```

**Model:**

**UserModel.php**

```
<?php
```

```
namespace App\Models;
```

```
use CodeIgniter\Model;
```

```
class UserModel extends Model
```

```
{
```

```
protected $table = 'user'; // name of your table
```

```
protected $primaryKey = 'user_id'; // primary key column
```

```
// columns you can insert or update
```

```
protected $allowedFields = ['username', 'email', 'password'];
```

```
// Auto-hash password before insert or update
```

```
protected $beforeInsert = ['hashPassword'];
```

```
protected $beforeUpdate = ['hashPassword'];
```

```
// 🔑 Automatically hash password
```

```
protected function hashPassword(array $data)
```

```
{
```

```
if (isset($data['data']['password'])) {
```

```
    $data['data']['password'] = password_hash($data['data']['password'], PASSWORD_DEFAULT);
```

```
}
```

```
return $data;
```

```
}
```

```
// 🔑 Find user by username for login
```

```
public function getUserByUsername(string $username)
```

```
{
```

```
    return $this->where('username', $username)->first();
```

```
}
```

```
}
```

**ProductModel.php**

```
<?php
```

```
namespace App\Models;
```

```

use CodeIgniter\Model;

class ProductModel extends Model
{
    protected $table      = 'products';
    protected $primaryKey = 'id';

    protected $allowedFields = ['product_name', 'description', 'price'];

    // Enable automatic timestamps
    protected $useTimestamps = true;

    // Define the column names in your DB
    protected $createdField = 'created_at';
    protected $updatedField = 'updated_at';

    // Helper methods for clarity (optional)
    public function createProduct($data)
    {
        return $this->insert($data);
    }

    public function getProduct($id)
    {
        return $this->find($id);
    }

    public function updateProduct($id, $data)
    {
        return $this->update($id, $data);
    }

    public function deleteProduct($id)
    {
        return $this->delete($id);
    }
}

```

## Controllers

### BaseController.php

```

<?php

namespace App\Controllers;

use CodeIgniter\Controller;
use CodeIgniter\HTTP\CLIRequest;
use CodeIgniter\HTTP\IncomingRequest;
use CodeIgniter\HTTP\RequestInterface;
use CodeIgniter\HTTP\ResponseInterface;
use Psr\Log\LoggerInterface;

/**
 * Class BaseController
 *
 * BaseController provides a convenient place for loading components
 * and performing functions that are needed by all your controllers.
 *
 * Extend this class in any new controllers:
 *
 *   class Home extends BaseController
 *
 * For security be sure to declare any new methods as protected or private.
 */
abstract class BaseController extends Controller
{
    /**
     * Instance of the main Request object.
     *
     * @var CLIRequest|IncomingRequest
     */
    protected $request;

    /**
     * An array of helpers to be loaded automatically upon
     * class instantiation. These helpers will be available
     * to all other controllers that extend BaseController.
     *
     * @var list<string>
     */
    protected $helpers = [];

}

```

```

 * Be sure to declare properties for any property fetch you initialized.

 * The creation of dynamic property is deprecated in PHP 8.2.

 */

// protected $session;

/**
 * @return void
 */

public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)
{
    // Do Not Edit This Line
    parent::initController($request, $response, $logger);

    // Preload any models, libraries, etc, here.

    // E.g.: $this->session = service('session');

}
}

```

### **Home.php**

```

<?php

namespace App\Controllers;

class Home extends BaseController
{
    public function index(): string
    {
        return view('welcome_message');
    }
}

```

### **PageController.php**

```

<?php

namespace App\Controllers;

class PageController extends BaseController
{
    public function pc()

```

```

{
    // Load products and pass to the product_card view
    $productModel = new \App\Models\ProductModel();
    $products = $productModel->findAll();

    return view('product_card', [
        'products' => $products,
    ]);
}

public function reports()
{
    // Load products and pass to the reports view
    $productModel = new \App\Models\ProductModel();
    $products = $productModel->findAll();

    return view('reports', [
        'products' => $products,
    ]);
}

```

### **ProductController.php**

```

<?php

namespace App\Controllers;

use App\Models\ProductModel;

class ProductController extends BaseController
{
    // Show all products with search and pagination
    public function index()
    {
        $productModel = new ProductModel();

        // Get search keyword from query string (?keyword=...)
        $keyword = $this->request->getGet('keyword');

        // Get selected currency (default USD)
        $currency = $this->request->getGet('currency') ?? 'USD';
    }
}

```

```

if ($keyword) {

    // Search by product_name or description with pagination
    $products = $productModel->like('product_name', $keyword)
        ->orLike('description', $keyword)
        ->paginate(5); // 5 items per page

    // Get total count for search results
    $totalCount = $productModel->like('product_name', $keyword)
        ->orLike('description', $keyword)
        ->countAllResults();

} else {

    $products = $productModel->paginate(5); // 5 items per page

    // Get total count of all products
    $totalCount = $productModel->countAll();

}

// Convert prices to selected currency
$exchangeRates = $this->getExchangeRates();

foreach ($products as &$product) {

    $product['original_price'] = $product['price'];

    $product['price'] = $this->convertCurrency($product['price'], 'USD', $currency, $exchangeRates);

    $product['currency_symbol'] = $this->getCurrencySymbol($currency);

}

$data = [
    'products' => $products,
    'pager'   => $productModel->pager,
    'totalCount' => $totalCount,
    'keyword'  => $keyword,
    'selectedCurrency' => $currency,
    'availableCurrencies' => ['USD', 'PHP', 'EUR', 'GBP', 'JPY', 'AUD', 'CAD']
];

return view('product_view', $data);
}

// Get exchange rates from API
private function getExchangeRates()

```

```

{

    // Check if rates are cached

    $cache = \Config\Services::cache();

    $cachedRates = $cache->get('exchange_rates');

    if ($cachedRates) {
        return $cachedRates;
    }

    // Fetch new rates

    try {

        $client = \Config\Services::curlrequest();

        $response = $client->get('https://api.exchangerate-api.com/v4/latest/USD', [
            'timeout' => 5
        ]);

        $data = json_decode($response->getBody(), true);

        $rates = $data['rates'] ?? [];

        // Cache for 1 hour

        $cache->save('exchange_rates', $rates, 3600);

        return $rates;
    } catch (\Exception $e) {

        // Return default rates if API fails

        return [
            'USD' => 1,
            'PHP' => 56.50,
            'EUR' => 0.92,
            'GBP' => 0.79,
            'JPY' => 149.50,
            'AUD' => 1.53,
            'CAD' => 1.36
        ];
    }
}

// Convert currency

private function convertCurrency($amount, $from, $to, $rates)

{

```

```

if ($from === $to) {
    return $amount;
}

// Convert from USD to target currency
if (isset($rates[$to])) {
    return round($amount * $rates[$to], 2);
}

return $amount;
}

// Get currency symbol
private function getCurrencySymbol($currency)
{
    $symbols = [
        'USD' => '$',
        'PHP' => '₱',
        'EUR' => '€',
        'GBP' => '£',
        'JPY' => '¥',
        'AUD' => 'A$',
        'CAD' => 'C$'
    ];

    return $symbols[$currency] ?? '$';
}

// Show create form

// Save product to DB
public function store()
{
    $productModel = new ProductModel();

    // Validation rules
    $validation = \Config\Services::validation();
    $validation->setRules([
        'product_name' => [
            'label' => 'Product Name',

```

```

'rules' => 'required|min_length[3]|max_length[100]|regex_match[/^A-Za-z0-9\s\-\_]+$/',
'errors' => [
    'required' => 'Product name is required',
    'min_length' => 'Product name must be at least 3 characters',
    'max_length' => 'Product name cannot exceed 100 characters',
    'regex_match' => 'Product name can only contain letters, numbers, spaces, hyphens, and underscores'
]
],
'description' => [
    'label' => 'Description',
    'rules' => 'permit_empty|min_length[10]|max_length[500]',
    'errors' => [
        'min_length' => 'Description must be at least 10 characters if provided',
        'max_length' => 'Description cannot exceed 500 characters'
    ]
],
'price' => [
    'label' => 'Price',
    'rules' => 'required|decimal|greater_than[0]|less_equal_to[999999.99]',
    'errors' => [
        'required' => 'Price is required',
        'decimal' => 'Price must be a valid decimal number',
        'greater_than' => 'Price must be greater than 0',
        'less_equal_to' => 'Price cannot exceed 999,999.99'
    ]
]
]);

```

```

if (!$validation->withRequest($this->request)->run()) {
    $errors = $validation->getErrors();
    $errorMessage = implode('<br>', $errors);
    session()->setflashdata('error', $errorMessage);
    return redirect()->back()->withInput();
}

$data = [
    'product_name' => trim($this->request->getPost('product_name')),
    'description' => trim($this->request->getPost('description')),
    'price'      => $this->request->getPost('price'),
];

```

```

if ($productModel->insert($data)) {
    session()->setFlashdata('message', 'Product added successfully!');
} else {
    session()->setFlashdata('error', 'Failed to add product. Please try again.');
}

return redirect()->to(base_url('/dashboard'));
}

// Update product
public function update($id)
{
    $productModel = new ProductModel();

    // Validation rules (same as store method)
    $validation = \Config\Services::validation();
    $validation->setRules([
        'product_name' => [
            'label' => 'Product Name',
            'rules' => 'required|min_length[3]|max_length[100]|regex_match/^([A-Za-z0-9\s\-\_]+)$/',
            'errors' => [
                'required' => 'Product name is required',
                'min_length' => 'Product name must be at least 3 characters',
                'max_length' => 'Product name cannot exceed 100 characters',
                'regex_match' => 'Product name can only contain letters, numbers, spaces, hyphens, and underscores'
            ]
        ],
        'description' => [
            'label' => 'Description',
            'rules' => 'permit_empty|min_length[10]|max_length[500]',
            'errors' => [
                'min_length' => 'Description must be at least 10 characters if provided',
                'max_length' => 'Description cannot exceed 500 characters'
            ]
        ],
        'price' => [
            'label' => 'Price',
            'rules' => 'required|decimal|greater_than[0]|less_equal_to[999999.99]',
            'errors' => [

```

```

'required' => 'Price is required',
'decimal' => 'Price must be a valid decimal number',
'greater_than' => 'Price must be greater than 0',
'less_than_equal_to' => 'Price cannot exceed 999,999.99'

]

]

]);


if (!$validation->withRequest($this->request)->run()) {

$errors = $validation->getErrors();

$errorMessage = implode('<br>', $errors);

session()->setflashdata('error', $errorMessage);

return redirect()->back()->withInput();

}






$data = [

'product_name' => trim($this->request->getPost('product_name')),

'description' => trim($this->request->getPost('description')),

'price'      => $this->request->getPost('price'),

];




if ($productModel->update($id, $data)) {

session()->setflashdata('message', 'Product updated successfully!');

} else {

session()->setflashdata('error', 'Failed to update product. Please try again.');

}






return redirect()->to(base_url('/dashboard'));

}






// Delete product

public function delete($id)

{

$productModel = new ProductModel();

$productModel->delete($id);





session()->setflashdata('message', 'Product deleted successfully!');

return redirect()->to(base_url('/dashboard'));

}

}

```

**AuthController.php**

```
<?php

namespace App\Controllers;

use App\Models\ProductModel;
use App\Models\UserModel;
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception as PHPMailerException;

class AuthController extends BaseController
{
    public function loginform()
    {
        return view('login');
    }

    public function register()
    {
        return view('register');
    }

    public function login()
    {
        $username = $this->request->getPost('username');
        $password = $this->request->getPost('password');

        $userModel = new UserModel();
        $user = $userModel->getUserByUsername($username);

        if ($user && password_verify($password, $user['password'])) {
            session()->set('user', [
                'user_id' => $user['user_id'],
                'username' => $user['username']
            ]);

            // ✅ Flashdata for SweetAlert
            return redirect()->to('/dashboard')->with('success', 'Welcome ' . $user['username']);
        } else {
            // ❌ Invalid credentials
            return redirect()->back()->with('error', 'Invalid username or password');
        }
    }
}
```

```

    }

}

public function registerUser()
{
    $userModel = new UserModel();

    // Validation rules
    $validation = \Config\Services::validation();
    $validation->setRules([
        'username' => [
            'label' => 'Username',
            'rules' => 'required|min_length[3]|max_length[50]|is_unique[user.username]',
            'errors' => [
                'required' => 'Username is required',
                'min_length' => 'Username must be at least 3 characters',
                'max_length' => 'Username cannot exceed 50 characters',
                'is_unique' => 'Username already exists'
            ]
        ],
        'email' => [
            'label' => 'Email',
            'rules' => 'required|valid_email|is_unique[user.email]',
            'errors' => [
                'required' => 'Email is required',
                'valid_email' => 'Please provide a valid email address',
                'is_unique' => 'Email already exists'
            ]
        ],
        'password' => [
            'label' => 'Password',
            'rules' => 'required|min_length[6]',
            'errors' => [
                'required' => 'Password is required',
                'min_length' => 'Password must be at least 6 characters'
            ]
        ]
    ]);
}

// ✅ Check if validation fails
if (!$this->validate($validation->getRules())) {

```

```

        return redirect()->back()->withInput()->with('errors', $validation->getErrors());
    }

    // Prepare data to save
    $data = [
        'username' => $this->request->getPost('username'),
        'email'    => $this->request->getPost('email'),
        'password' => $this->request->getPost('password')
    ];

    // Try saving and surface DB/model errors back to the user
    try {
        $saved = $userModel->save($data);

        if ($saved === false) {
            // Collect model errors if any
            $errors = $userModel->errors();

            $msg = is_array($errors) ? implode(' ', $errors) : 'Unable to save user.';

            return redirect()->back()->withInput()->with('error', $msg);
        }
    } catch (\Throwable $e) {
        // Log the exception for debugging and return a friendly message
        if (function_exists('log_message')) {
            log_message('error', 'User save failed: ' . $e->getMessage());
        }

        return redirect()->back()->withInput()->with('error', 'Registration failed: database error. Check logs.');
    }

    // ✅ Redirect to login with a flash message
    return redirect()->to('/')->with('success', 'Registration successful! Please log in.');
}

public function logout()
{
    session()->destroy();      // If the request includes ?silent=1 we skip setting a flash message
    $silent = $this->request->getGet('silent');

    if ($silent === '1') {
        return redirect()->to('/');
    }

    return redirect()->to('/')->with('success', 'Logged Out Successfully');
}

```

```

// Show forgot password form

public function forgotForm()
{
    return view('auth/forgot_password');
}

// Handle form: create signed token (no DB) and send reset email

public function sendReset()
{
    $email = $this->request->getPost('email');

    if (empty($email)) {
        return redirect()->back()->with('error', 'Please provide your email.');
    }

    $userModel = new UserModel();
    $user = $userModel->where('email', $email)->first();

    if (!$user) {
        return redirect()->back()->with('error', 'Email not found.');
    }

    // create signed token: userId|expires|hmac
    $userId = $user[$userModel->primaryKey] ?? $user['user_id'] ?? null;
    $expires = time() + 3600; // 1 hour
    $payload = $userId . '|' . $expires;

    $appConfig = config('App');
    $key = $appConfig->encryptionKey ?: getenv('APP_KEY') ?: 'change_this_key';
    $sig = hash_hmac('sha256', $payload, $key);
    $token = rtrim(strtr(base64_encode($payload . '|' . $sig), '+/_', '-_'), '=');

    $link = base_url("reset-password?token={$token}");

    // Try using PHPMailer (Composer) if available, otherwise fall back to CodeIgniter Email

    $emailConfig = config('Email');
    $sent = false;

    if (class_exists(\PHPMailer\PHPMailer\PHPMailer::class)) {
        try {
            $mail = new PHPMailer(true);

```

```

$mail->isSMTP();

$mail->Host      = 'smtp.gmail.com';

$mail->SMTPAuth  = true;

$mail->Username   = 'von.vergara.399@gmail.com';    // your Gmail

$mail->Password   = 'ohdh dksv zyvd ihsn';        // Gmail App Password

$mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS; // TLS

$mail->Port       = 587;

$fromEmail = $emailConfig->fromEmail ?: 'no-reply@example.local';

$fromName = $emailConfig->fromName ?: 'SI-CRUD';

$mail->setFrom($fromEmail, $fromName);

$mail->addAddress($email);

$mail->isHTML(true);

$mail->Subject = 'Password reset request';

$mail->Body   = view('emails/reset_password', ['link' => $link, 'expires' => 60]);

$mail->send();

$sent = true;

} catch (PHPMailerException $e) {

    log_message('error', 'PHPMailer exception: ' . $e->getMessage());

    // fallthrough to CI mailer

    $sent = false;

} catch (\Throwable $e) {

    log_message('error', 'PHPMailer error: ' . $e->getMessage());

    $sent = false;

}

}

if (!$sent) {

    // Fallback to CodeIgniter Email service (existing behavior)

    $emailer = \Config\Services::email();

    $from = $emailConfig->fromEmail ?: 'no-reply@example.local';

    $fromName = $emailConfig->fromName ?: 'SI-CRUD';

    $emailer->setFrom($from, $fromName);

    $emailer->setTo($email);

    $emailer->setSubject('Password reset request');

    $message = view('emails/reset_password', ['link' => $link, 'expires' => 60]);

    $emailer->setMessage($message);

}

if (!$emailer->send()) {

```

```

    log_message('error', 'Reset email failed: ' . $emailer->printDebugger(['headers']));
    return redirect()->back()->with('error', 'Unable to send reset email. Try again later.');
}

}

return redirect()->to('/')->with('success', 'Reset link sent. Check your email.');
}

// Show reset form (token in query)

public function resetForm()
{
    $token = $this->request->getGet('token');

    if (!$token) {
        return redirect()->to('/')->with('error', 'Invalid reset link.');
    }

    $valid = $this->validateResetToken($token);

    if ($valid === false) {
        return redirect()->to('/')->with('error', 'Reset link expired or invalid.');
    }

}

// pass token to view (hidden input)

return view('auth/reset_password', ['token' => $token]);
}

// Accept new password and update user

public function resetPassword()
{
    $token = $this->request->getPost('token');

    $password = $this->request->getPost('password');

    $password_confirm = $this->request->getPost('password_confirm');

    if (!$token) {
        return redirect()->to('/')->with('error', 'Invalid request.');
    }

    $valid = $this->validateResetToken($token);

    if ($valid === false) {
        return redirect()->to('/')->with('error', 'Reset link expired or invalid.');
    }
}

```

```

// basic validation

if (empty($password) || strlen($password) < 6 || $password !== $password_confirm) {
    return redirect()->back()->withInput()->with('error', 'Passwords must match and be at least 6 characters.');
}

$userModel = new UserModel();

// 💡 DO NOT hash here, model will auto-hash

$updated = $userModel->update($valid['user_id'], [
    'password' => $password
]);

if ($updated === false) {
    return redirect()->back()->with('error', 'Unable to update password. Try again.');
}

return redirect()->to('/')->with('success', 'Password updated. Please login.');
}

// Helper: validate token and return ['user_id'=>.., 'expires'=>..] or false

private function validateResetToken(string $token)
{
    $decoded = base64_decode(str_replace(['.', '_'], ['+', '/'], $token));
    if (!$decoded) {
        return false;
    }

    $parts = explode('|', $decoded);
    if (count($parts) !== 3) {
        return false;
    }

    list($userId, $expires, $sig) = $parts;
    if ((int)$expires < time()) {
        return false;
    }
}

	payload = $userId . '|' . $expires;
$appConfig = config('App');
$key = $appConfig->encryptionKey ?: getenv('APP_KEY') ?: 'change_this_key';
$expected = hash_hmac('sha256', $payload, $key);

```

```
if (!hash_equals($expected, $sig)) {  
    return false;  
}  
return ['user_id' => $userId, 'expires' => $expires];  
}  
}
```

### **Routes.php**

```
<?php

use CodeIgniter\Commands\Utilities\Routes;
use CodeIgniter\Router\RouteCollection;

/**
 * @var RouteCollection $routes
 */

// Homepage route

$routes->get('/', 'AuthController::loginform'); // Show create view
$routes->get('/register', 'AuthController::register'); // Show create view
$routes->post('/register', 'AuthController::registerUser'); // Handle registration
$routes->post('/login', 'AuthController::login'); // Handle login
$routes->get('/logout', 'AuthController::logout'); // Handle logout

$routes->get('forgot-password', 'AuthController::forgotForm');
$routes->post('forgot-password', 'AuthController::sendReset');
$routes->get('reset-password', 'AuthController::resetForm');
$routes->post('reset-password', 'AuthController::resetPassword');

$routes->get('/dashboard', 'ProductController::index'); // Show all products
$routes->post('products/store', 'ProductController::store');
$routes->post('products/update/(:num)', 'ProductController::update/$1');
$routes->post('products/delete/(:num)', 'ProductController::delete/$1');

$routes->get('/product_card', 'PageController::pc'); // Show product card view
$routes->get('/reports', 'PageController::reports'); // Show product card view
```

### **Video Link:**

<https://www.facebook.com/reel/826110336997602>