

## **Admin Registration (admin\_register.php)**

This PHP script handles the secure registration of new admin accounts. When accessed via a POST request, it retrieves and sanitizes the submitted username and password values. If any required fields are empty, the user is redirected back to the registration form with an error message. The script checks the admins table to ensure the chosen username is unique, preventing duplicate accounts. If the username is available, it securely hashes the password using `password_hash()` and inserts the new admin record into the database. Upon successful registration, a success message is stored in the session; otherwise, an appropriate error message is set. In all cases, the user is redirected back to `admin_register_form.php`. This script is essential for controlled creation of administrative users, ensuring both security and data integrity.

## **Username & Password (login.php)**

This PHP script handles admin authentication for the dashboard by verifying login credentials submitted via a POST request. It begins by starting a session, validating that the request method is POST, and loading database configuration from an external file. Instead of relying on hardcoded credentials, the script now queries a database table (admins) to retrieve the corresponding user record based on the submitted username. It securely verifies the password using `password_verify()` against the hashed password stored in the database. Upon successful authentication, session variables are set and the user is redirected to `dashboard.php`. If the credentials are invalid or missing, an appropriate error message is stored in the session and the user is redirected back to the login form. This update improves security and scalability by supporting database-driven user management.

## **Admin Dashboard (dashboard.php)**

The Admin Dashboard provides the website administrator with a concise overview of essential metrics, including the total number of products and orders within the system. It first ensures secure access by verifying if the admin is logged in; unauthenticated users are redirected to the login page. Using PHP sessions, the dashboard maintains secure access control. The page retrieves data from the database using PDO queries to count total products and total orders, displaying these counts clearly. It also includes basic error handling—if database queries fail, the counts are safely defaulted to zero to maintain page stability. The database connection details are maintained in a central configuration file (`../includes/config.php`), promoting modularity and easier maintenance. Lastly, the visual presentation and layout for this page are handled separately via the included template (`../admin/admin_dashboard_template.php`).

## **Add Product Page (add\_product.php)**

The "Add Product" page allows administrators to easily insert new products into the website's inventory through a user-friendly form. Administrators can input essential product details such as product name, description, price, and upload an image file directly from their local system. The page ensures secure form handling with the use of POST method and proper file upload management. All required fields are clearly marked and validated client-side to enhance usability and data integrity. Additionally, the page is styled with an intuitive and

responsive design, providing clarity and ease of use. Upon submission, form data is sent securely to the backend (add\_product.php processing script), which handles database insertion. The layout and styling rely on an external CSS file (./css/style.css) and inline CSS for specific admin-form aesthetics. Header (admin\_header.php) and footer (footer.php) templates are included separately, ensuring consistency across administrative pages.

### **Edit Product Page (edit\_product.php)**

The Edit Product page enables administrators to securely and efficiently update existing product details within the system. It begins by validating the administrator's session, ensuring only authorized users can perform edits. Administrators can update key product information, including product name, description, price, and image. The current product details are retrieved from the database and pre-filled into the form fields, streamlining the editing process. The form handles both textual data and image uploads; if a new image isn't uploaded, the existing product image remains unchanged. Upon submission, the page updates the database securely using prepared statements via PDO. The page provides clear feedback messages indicating success or failure of the update operation. User-friendly inline validation, file-upload handling, and responsive styling improve usability. The page integrates a consistent look and feel through external CSS (./css/style.css) and includes reusable header and footer templates (admin\_header.php and footer.php).

### **Admin - Answer Customer Questions (admin\_questions.php)**

The "Answer Customer Questions" page provides administrators with a centralized interface to view, manage, and respond directly to customer inquiries. The page securely ensures that only logged-in administrators can access the question-answering functionality. It retrieves all customer-submitted questions from the database, sorting them by date, displaying clearly each question's subject, content, and submission timestamp. If a question has not yet been answered, the admin can provide an answer through an intuitive inline form, updating the database directly with the response. Already-answered questions display the provided answer prominently. The page provides immediate feedback to confirm successful updates or notify administrators in case of errors. Its user-friendly design leverages external CSS (./css/style.css) alongside customized inline styling for clear readability. Standard header (admin\_header.php) and footer templates (footer.php) are included to ensure consistency across the administrative interface.

### **Admin - RMA Requests (admin\_rma.php)**

The Admin RMA Requests page enables administrators to efficiently review and respond to customer-submitted Return Merchandise Authorization (RMA) requests. This page securely checks for authenticated admin access and retrieves all RMA requests from the database, displaying them in reverse chronological order. Each RMA request clearly shows key details, including the request ID, member ID, related order item IDs, customer's question or request, the date of submission, and any documents attached by the customer. Administrators can directly provide responses through a convenient form embedded alongside each request. The form submission updates the database securely using PDO prepared statements, providing immediate feedback messages confirming successful updates or highlighting any issues. The page maintains

consistent design standards through external CSS (../css/style.css) combined with page-specific inline CSS, and utilizes common header (admin\_header.php) and footer (footer.php) templates for a cohesive administrative experience.

### **Admin - Manage Stock (admin\_stock.php)**

The Manage Stock page allows administrators to easily review and update product inventory levels within a straightforward, intuitive interface. It begins by securely verifying that the admin is logged in. Once authenticated, the administrator sees a complete overview of all products, including each product's ID, name, current stock quantity, and a field for entering updated stock levels. The admin can conveniently enter new stock quantities directly within a structured table, facilitating quick bulk updates. Upon submission, the page securely updates the database using PDO prepared statements, either inserting new stock records or updating existing ones seamlessly. Administrators receive clear, immediate feedback confirming successful updates or highlighting errors. The user interface leverages external CSS (../css/style.css) along with customized inline styles to ensure readability and ease of use. The consistent layout includes standard header (admin\_header.php) and footer templates (footer.php) across administrative pages.

### **Admin - Website Monitoring (admin\_monitor.php)**

The Website Monitoring page provides administrators with a straightforward, real-time overview of the availability and responsiveness of critical web pages within the website. This page checks several key pages, such as the Home Page, Search Page, Cart, and Membership/Login, using HTTP requests to verify their operational status. Each monitored service is clearly displayed in a table format indicating whether it is currently "Online" or "Offline." The page utilizes PHP's cURL functionality to perform these checks, interpreting HTTP response codes to determine the status. Administrators can quickly identify issues, allowing for prompt troubleshooting. Authentication ensures secure access, restricting monitoring capabilities exclusively to logged-in admins. The interface design incorporates external CSS (css/style.css) and inline styling for clarity and usability, maintaining consistency with other administrative pages through the inclusion of standard header (admin\_header.php) and footer templates.

### **Admin - Template Switcher (admin\_template.php)**

The Admin Template Switcher page allows website administrators to dynamically select and apply visual themes or templates to the entire site effortlessly. It securely validates administrator access, ensuring only authorized users can change site appearance. Administrators can choose from predefined template options (Default, Blue, and Red) presented via clear and intuitive buttons. Upon selection, the chosen template setting is stored securely in a cookie for 30 days, immediately applying the visual changes site-wide. The interface prominently displays the currently active template, providing instant visual confirmation. Design consistency is maintained through external CSS (../css/style.css) and additional inline CSS for specific layout elements, alongside standard header (admin\_header.php) and footer (footer.php) templates for cohesive navigation.

## **Admin - Account Administration (admin\_account\_administration.php)**

The Account Administration page allows administrators to securely manage user accounts within the website, providing comprehensive control over member access and activity monitoring. It ensures only authenticated administrators can view or modify account statuses. Administrators can view all registered members displayed clearly in a structured table showing member ID, username, email, and current account status. They can easily enable or disable user accounts through intuitive action buttons that securely update the database. Additionally, each user account features an expandable history section that administrators can toggle, displaying detailed login/logout events, including timestamps and IP addresses. The interface uses a combination of external CSS (`../css/style.css`) and inline styles to provide a clean and responsive layout. JavaScript enhances interactivity, enabling administrators to show or hide user history dynamically without page reloads. Consistent layout and navigation are maintained by incorporating standard header (`admin_header.php`) and footer (`footer.php`) templates.