**Report on Secure Login System Implementation**

**Introduction**

This report provides an overview of the secure login system developed using HTML, JavaScript, and PHP. The system ensures secure user authentication by implementing password hashing, session-based authentication, and SQL injection prevention mechanisms.

**System Overview**

The login system consists of the following key components:

1. **Database Connection (db.php)**
2. **User Registration (register.php)**
3. **User Login (login.php)**
4. **User Dashboard (dashboard.php)**
5. **User Logout (logout.php)**
6. **Frontend (login.html)**

**Explanation of Components**

**1. Database Connection (db.php)**

This file establishes a secure connection to a MySQL database using **PDO (PHP Data Objects)**. PDO provides a secure way to interact with databases and supports prepared statements, which protect against SQL injection attacks.

**2. User Registration (register.php)**

* Accepts a username and password from the user.
* Uses password\_hash() with the **bcrypt algorithm** to securely store passwords.
* Stores the hashed password in the database.

**3. User Login (login.php)**

* Accepts login credentials from the user.
* Retrieves the stored hashed password using a **prepared statement** to prevent SQL injection.
* Uses password\_verify() to compare the entered password with the stored hash.
* If authentication is successful, the system starts a **session** and sets an **HTTP-only cookie** for security.

**4. User Dashboard (dashboard.php)**

* Displays a personalized welcome message.
* Ensures that only authenticated users can access the dashboard by checking for an active session.
* If no valid session is detected, redirects the user to the login page.

**5. User Logout (logout.php)**

* Destroys the session to log out the user.
* Clears authentication cookies to prevent unauthorized access.
* Redirects the user back to the login page.

**6. Frontend (login.html)**

* Provides a simple login form for user authentication.
* The form submits login details to login.php using the **POST method**, ensuring that sensitive information is not exposed in the URL.

**Security Features Implemented**

* **Session-based Authentication**: Maintains user login state securely.
* **Password Hashing**: Uses password\_hash() to store encrypted passwords.
* **SQL Injection Prevention**: Uses **prepared statements** to protect database queries.
* **Secure Cookies**: Sets authentication cookies with **HTTP-only** and **secure** flags to prevent attacks.

**Conclusion**

This secure login system follows best practices in web security, including password hashing, session management, and SQL injection prevention. It ensures that user data remains protected while providing a seamless authentication experience.