**authentication and authorization**

تعد المصادقة والترخيص عمليتين حيويتين لأمن المعلومات يستخدمهما المسؤولون لحماية الأنظمة والمعلومات. تتحقق المصادقة من هوية المستخدم أو الخدمة، ويحدد الترخيص حقوق الوصول الخاصة بهم.

**Authentication**:

المصادقة هي عملية التحقق من هوية المستخدم أو النظام. إنه يجيب على السؤال "من أنت؟" إنه يضمن أن المستخدم هو من يدعي أنه. في PHP، تتضمن المصادقة عادةً التحقق من بيانات اعتماد المستخدم (على سبيل المثال، اسم المستخدم وكلمة المرور) مقابل سجل مخزن في قاعدة بيانات أو أي نظام مصادقة آخر.

تتضمن طرق المصادقة الشائعة في PHP ما يلي:

مصادقة اسم المستخدم وكلمة المرور: يقدم المستخدمون اسم مستخدم وكلمة مرور، والتي تتم مقارنتها ببيانات الاعتماد المخزنة لمنح حق الوصول.

المصادقة المستندة إلى الرمز المميز: يتلقى المستخدمون رمزًا مميزًا بعد تسجيل الدخول بنجاح، والذي يتم استخدامه بعد ذلك للطلبات اللاحقة.

OAuth وOpenID Connect: هذه هي البروتوكولات المستخدمة للمصادقة في أنظمة الجهات الخارجية، مثل تسجيلات الدخول إلى الوسائط الاجتماعية.

المصادقة البيومترية: استخدام بصمة الإصبع أو التعرف على الوجه لتحديد هوية المستخدم

**Authorization**:

من ناحية أخرى، التفويض هو عملية تحديد ما إذا كان المستخدم أو النظام لديه الأذونات والامتيازات اللازمة للوصول إلى مورد معين أو تنفيذ إجراء محدد. إنه يجيب على السؤال: "ما الذي يُسمح لك بفعله؟" يتحكم التفويض في ما يمكن للمستخدم أو لا يمكنه فعله داخل التطبيق بعد المصادقة عليه.

في PHP، يتم تنفيذ التفويض عادةً عن طريق التحقق من أدوار المستخدم أو أذوناته أو سماته مقابل الأذونات المطلوبة لإجراء أو مورد معين.

تتضمن طرق الترخيص الشائعة في PHP ما يلي:

التحكم في الوصول المستند إلى الأدوار (RBAC): يتم تعيين أدوار للمستخدمين (على سبيل المثال، المسؤول والمستخدم) وترتبط الأذونات بهذه الأدوار.

التحكم في الوصول المستند إلى السمات (ABAC): تعتمد قرارات الوصول على سمات مختلفة، مثل سمات المستخدم وسمات الموارد.

قوائم التحكم في الوصول (ACLs): قوائم تحدد الإجراءات أو الموارد التي يمكن للمستخدم أو الدور الوصول إليها.

البرامج الوسيطة: غالبًا ما تستخدم أطر عمل PHP البرامج الوسيطة لفرض قواعد الترخيص في نقاط مختلفة في دورة حياة الطلب/الاستجابة للتطبيق

لتلخيص ذلك، تتعلق المصادقة بالتحقق من هوية المستخدم، بينما يتعلق التفويض بتحديد الإجراءات أو الموارد التي يُسمح للمستخدم بالوصول إليها أو تنفيذها بعد المصادقة. كلاهما ضروري لضمان الأمان والتحكم في الوصول إلى تطبيق الويب PHP الخاص بك.

**Post vs Get**

استخدم GET لاسترداد البيانات والعمليات البسيطة وغير الفعالة،

واستخدم POST لإرسال البيانات إلى الخادم، خاصة للعمليات التي تعدل البيانات أو تتضمن معلومات حساسة.

**Filter\_var**

filter\_var هي وظيفة PHP مدمجة تُستخدم لتصفية البيانات والتحقق من صحتها، وخاصة مدخلات المستخدم. ويشيع استخدامه لتطهير مدخلات المستخدم والتحقق من صحتها، مثل البيانات من نماذج HTML، لمنع الثغرات الأمنية وضمان سلامة البيانات.

**Session unset vs Session destroy**

session\_unset()

الغرض: يتم استخدام session\_unset() لمسح كافة المتغيرات المخزنة في الجلسة الحالية. فهو يزيل جميع البيانات المرتبطة بالجلسة بشكل فعال دون تدمير الجلسة نفسها.

التأثير: يجعل متغيرات الجلسة فارغة ولكنه يبقي الجلسة نفسها حية. يمكنك الاستمرار في استخدام الجلسة عن طريق تعيين قيم جديدة لمتغيرات الجلسة بعد استدعاء session\_unset().

session\_destroy()

الغرض: يتم استخدام session\_destroy() لإنهاء الجلسة وتدميرها بالكامل، بما في ذلك بيانات الجلسة ومعرف الجلسة. غالبًا ما يتم استخدامه لتسجيل خروج المستخدم أو إنهاء جلسة المستخدم تمامًا.

التأثير: ينهي الجلسة، ويمسح جميع بيانات الجلسة، وينشئ معرف جلسة جديد للجلسة اللاحقة إذا بدأت واحدة. بعد استدعاء session\_destroy()، تحتاج عادةً إلى استدعاء session\_start() لبدء جلسة جديدة.

**html\_entities vs htmlspecialchar**

htmlentities وhtmlspecialchars هي وظائف تستخدم للهروب وترميز الأحرف الخاصة في سلسلة لمخرجات HTML لمنع هجمات البرمجة النصية عبر المواقع (XSS) والتأكد من عرض HTML بشكل صحيح. ومع ذلك، فهي تختلف في سلوك الترميز الخاص بها:

htmlentities

الغرض: يتم استخدام htmlentities بشكل أساسي لترميز الأحرف ككيانات HTML. يقوم بتحويل الأحرف الخاصة إلى تمثيلات كيان HTML المقابلة لها. على سبيل المثال، يقوم بتحويل < إلى &lt;، > إلى &gt;، & إلى &amp;، وهكذا.

مستوى الترميز: تقوم htmlentities بتشفير الأحرف إلى نموذج كيان HTML الخاص بها لجميع الأحرف ذات كيانات HTML المقابلة، بما في ذلك الأحرف غير ASCII مثل الحروف المحركة. وهذا يجعلها أكثر ملاءمة للتدويل.

htmlspecialchars

الغرض: يتم استخدام htmlspecialchars لترميز الأحرف ككيانات HTML، ولكنه يركز على مجموعة فرعية من الأحرف الأكثر صلة بمنع هجمات XSS. يقوم بتحويل الأحرف مثل < و> و& و" و" إلى تمثيلات كيان HTML المقابلة لها.

مستوى التشفير: تم تصميم htmlspecialchars لتشفير الأحرف التي لها معنى خاص في HTML فقط، مما يجعله خيارًا خفيفًا أكثر للهروب من HTML الأساسي.

في معظم الحالات، يجب عليك استخدام htmlspecialchars لتلبية احتياجات الهروب الأساسية من HTML، خاصة عندما تريد منع هجمات XSS عن طريق ترميز الأحرف التي يمكن استخدامها لإدخال البرنامج النصي. إذا كانت لديك متطلبات محددة للتعامل مع أحرف غير ASCII أو كنت بحاجة إلى تشفير كافة الأحرف إلى كيانات HTML الخاصة بها، فيمكنك استخدام htmlentities. يعتمد الاختيار بينهما على مستوى التشفير والأحرف المحددة التي تريد تشفيرها لحالة الاستخدام الخاصة بك.