Portail web institutionnel dynamique et sécurisé pour ISTM Nyankunde

Étude de conception appliquée en informatique

Etsora Obhiza Rehema, Mumbere Nguramo Augustin
Email : etsoraracham@gmail.com

Mots-clés

Architecture web ; contrôle d'accès par rôles ; MySQL ; PHP ; portail universitaire ; publication scientifique ; sécurité des applications web ; traçabilité

RÉSUMÉ

Les établissements d'enseignement supérieur doivent publier des informations académiques, recevoir des candidatures et diffuser des travaux scientifiques sans exposer leurs données ni accorder des privilèges excessifs aux gestionnaires du site. Cette étude décrit la conception d'un portail web dynamique pour l'ISTM Nyankunde et examine la manière dont les exigences éditoriales ont été traduites en contrôles techniques. La démarche suit une méthode de recherche orientée conception. Elle part des besoins institutionnels, construit un artefact en PHP et MySQL, puis vérifie sa conformité fonctionnelle par l'inspection du code, du schéma de données et des parcours d'administration. Le portail sépare les responsabilités entre quatre rôles, impose un circuit de validation des contenus, protège les formulaires par jeton de session, utilise des requêtes préparées et contrôle les fichiers téléversés par type MIME et taille. Les publications scientifiques sont rattachées à une revue et à un numéro avant leur mise en ligne. Le téléchargement de chaque article peut être autorisé séparément ; lorsqu'il est désactivé, un lecteur en canvas limite l'interface publique aux quatre premières pages et neutralise les commandes usuelles d'impression et d'enregistrement. Les consultations, téléchargements et actions administratives sont journalisés. L'analyse montre qu'une architecture légère, compatible avec XAMPP, peut couvrir les besoins essentiels d'un institut tout en améliorant la traçabilité. Elle montre aussi que les restrictions appliquées dans le navigateur ne constituent pas une gestion numérique des droits et doivent être complétées par des contrôles serveur, des sauvegardes testées et une surveillance opérationnelle.

Keywords

academic portal; application security; audit logging; MySQL; PHP; Codex; role-based access control; scientific publishing; web architecture

Institut Supérieur des Techniques Médicales de Nyankunde, Bunia, République démocratique du Congo

ABSTRACT

Higher education institutions need to publish academic information, receive applications, and disseminate research without exposing sensitive data or granting excessive privileges to website managers. This study describes the design of a dynamic institutional portal for ISTM Nyankunde and examines how editorial requirements were translated into technical controls. The work follows a design science approach based on institutional needs, the construction of a PHP and MySQL artifact, and a functional verification of its source code, data model, and administrative workflows. The portal separates responsibilities across four roles, enforces content review before publication, protects forms with session tokens, uses prepared database statements, and validates uploaded files by MIME type and size. Scientific articles must be attached to an existing journal issue. Download permission is configured for each article; when disabled, a canvas reader exposes only the first four pages in the public interface and blocks common print and save commands. Page views, downloads, and administrative actions are recorded. The analysis indicates that a lightweight architecture compatible with XAMPP can satisfy the core operational needs of a small higher education institution while improving accountability. Browser-side restrictions remain a deterrent rather than digital rights management and therefore require complementary server controls, tested backups, and operational monitoring.

INTRODUCTION

Un site institutionnel d'enseignement supérieur ne se réduit plus à une vitrine. Il sert à publier des horaires, présenter les filières, recevoir des candidatures, diffuser des actualités et rendre visibles les activités de recherche. Dès que ces opérations sont administrées en ligne, le portail devient un système d'information. Il traite des identités, des documents et des décisions de publication. Une erreur d'autorisation peut alors avoir autant d'effet qu'une défaillance d'interface : un contenu non validé peut être publié, un dossier d'inscription peut être exposé ou une action administrative peut ne laisser aucune trace.

Le projet étudié répond à cette situation dans le contexte de l'Institut Supérieur des Techniques Médicales de Nyankunde. L'objectif était de remplacer un site difficile à administrer par une application dynamique bilingue, exploitable sur une installation XAMPP et organisée autour des activités réelles de l'établissement. La question de recherche est la suivante : comment concevoir un portail universitaire léger qui concilie administration déléguée, publication scientifique, protection des données et traçabilité des opérations ?

La contribution de cette étude est un retour de conception vérifiable. Elle décrit les choix d'architecture, relie les exigences aux mécanismes présents dans le code et identifie les limites qui subsistent avant une exploitation à grande échelle. Elle ne présente ni test d'intrusion certifié, ni mesure de charge, ni enquête de satisfaction. Les résultats portent sur l'artefact construit et sur la couverture observable de ses exigences.

PROBLÈME DE RECHERCHE ET OBJECTIFS

Le cahier des charges associait des fonctions éditoriales et des fonctions sensibles. Le même système devait administrer la page d'accueil, les filières, la valve hebdomadaire, le comité de gestion, les secteurs et bureaux, les actualités, les médias, les chercheurs, les revues, les articles et les candidatures. Les contenus devaient être proposés par plusieurs niveaux de comptes sans donner à chacun les mêmes droits. Les documents scientifiques devaient rester lisibles en ligne, avec une autorisation de téléchargement décidée article par article. Enfin, les responsables devaient disposer de statistiques, de notifications, d'un journal d'actions et d'un mécanisme de sauvegarde.

Trois objectifs ont guidé la conception. Le premier consistait à structurer les contenus et leurs relations afin qu'une revue et son numéro existent avant l'ajout d'un article. Le deuxième visait à appliquer le principe du moindre privilège dans l'espace d'administration. Le troisième portait sur la réduction des risques courants d'une application PHP : injection SQL, falsification de requête, téléversement dangereux, vol de session, exposition de fichiers et absence de journalisation. L'OWASP ASVS fournit une base de vérification des contrôles techniques d'une application web [2]. Le modèle RBAC du NIST fournit, quant à lui, un vocabulaire pour relier utilisateurs, rôles et permissions [3].

CADRE CONCEPTUEL

Recherche orientée conception

La recherche orientée conception étudie un problème par la construction et l'évaluation d'un artefact. La méthode proposée par Peffers et ses collègues enchaîne l'identification du problème, la définition des objectifs, la conception, la démonstration, l'évaluation et la communication des résultats [1]. Cette logique convient au présent travail, car l'objet étudié est une application opérationnelle et non une hypothèse détachée de son contexte d'usage.

Sécurité en profondeur

Aucun contrôle isolé ne suffit à sécuriser un portail. Les requêtes préparées réduisent le risque d'injection, mais elles ne remplacent ni l'autorisation, ni l'échappement des sorties, ni la protection de session. De la même manière, la validation d'un type MIME ne suffit pas à garantir qu'un fichier est inoffensif. L'OWASP recommande une liste de formats autorisés, des noms générés par l'application, des limites de taille, un stockage adapté et une autorisation explicite des personnes qui téléversent [7]. L'architecture étudiée a donc été évaluée comme un ensemble de couches complémentaires.

Contrôle des accès et traçabilité

Le modèle RBAC attribue des permissions à des rôles plutôt qu'à chaque utilisateur pris séparément. Cette séparation facilite l'application du moindre privilège et la révision des droits [3]. Dans un système éditorial, elle doit être complétée par un état du contenu. Le droit de créer ne doit pas impliquer automatiquement le droit de publier. La journalisation applicative complète ce mécanisme en enregistrant l'auteur, l'action, l'objet concerné et la date. L'OWASP souligne que les journaux applicatifs offrent une visibilité que les seuls journaux du serveur web ne fournissent pas [8].

MÉTHODE

L'étude a utilisé une démarche de conception appliquée inspirée du processus DSRM [1]. L'unité d'analyse est le portail dynamique de l'ISTM Nyankunde. Les données d'étude sont constituées du cahier des charges, du schéma relationnel MySQL, des contrôleurs PHP, des vues publiques et administratives, des règles de réécriture Apache, ainsi que des scripts JavaScript du lecteur de documents.

L'évaluation a suivi quatre opérations. Les besoins ont d'abord été regroupés par domaine fonctionnel. Chaque besoin a ensuite été relié à une table, une règle d'autorisation ou un parcours de l'application. Une inspection statique a vérifié les points de sécurité visibles dans le code : requêtes préparées, échappement HTML, jetons CSRF, paramètres de session, contrôle des téléversements, en-têtes HTTP et journalisation. Enfin, les parcours principaux ont été vérifiés fonctionnellement, notamment la création puis la soumission d'un contenu, la publication par un rôle autorisé, l'affichage d'un article protégé et l'activation de son téléchargement.

Cette méthode vérifie la cohérence interne de l'artefact, mais ne remplace pas un audit indépendant. Aucune conclusion n'est tirée sur la résistance à une attaque inconnue, la performance sous forte charge ou l'utilisabilité auprès d'un échantillon d'étudiants et d'agents.

ARCHITECTURE DU PORTAIL

Le portail suit une architecture en couches compatible avec PHP 8 et MySQL. La couche de présentation regroupe les vues HTML, les feuilles de style et les interactions JavaScript. La couche applicative contient les contrôleurs publics et administratifs, l'authentification, la sécurité des entrées, la gestion des sauvegardes et les fonctions de notification. La couche de données utilise PDO avec les requêtes préparées natives ; l'émulation des requêtes préparées est désactivée. Les adresses publiques sont réécrites par Apache afin de séparer les parcours lisibles des fichiers internes.

Tableau 1 — Principaux domaines de données du portail

Domaine	Entités principales	Finalité
Identité et gouvernance	users, settings, audit_logs, notifications	Comptes, rôles, paramètres, preuves d'action et alertes
Édition	contents, content_media, homepage_sections	Contenus bilingues, médias, états de publication et accueil
Recherche	journal_issues, researchers, contents	Revues, numéros, DOI, chercheurs et articles
Enseignement	programs, departments, schedules	Filières, orientations, auditoires et horaires
Admission	admission_applications, admission_documents	Candidatures, suivi de statut et pièces privées
Mesure et continuité	page_views, download_events, backups	Audience, téléchargements et sauvegardes

Gestion des rôles et du cycle éditorial

Quatre rôles structurent l'espace d'administration. Leur hiérarchie correspond à la responsabilité institutionnelle, mais les permissions sont contrôlées pour chaque opération. Le Owner possède tous les droits. Le Super Admin gère les contenus, les comptes autorisés, les paramètres, les candidatures et les sauvegardes. L'Admin crée des actualités, modifie uniquement ses propres contenus et les soumet à validation. Le User consulte le tableau de bord, les statistiques et ses notifications sans modifier les données.

Tableau 2 — Répartition des responsabilités administratives

Rôle	Création et modification	Publication et suppression	Fonctions sensibles
Owner	Tous les contenus et tous les comptes	Autorisé	Paramètres, sauvegardes, admissions et audit complet
Super Admin	Tous les contenus et	Autorisé	Paramètres, sauvegardes,

	comptes Admin ou User		admissions et audit
Admin	Actualités personnelles	Soumission seulement	Aucune suppression ni configuration système
User	Aucune	Aucune	Consultation du tableau de bord et du profil

Le cycle éditorial sépare le brouillon, la soumission et la publication. Lorsqu'un Admin soumet son contenu, le statut passe à pending et les responsables reçoivent une notification. Seuls le Owner et le Super Admin peuvent publier, retirer ou supprimer. Chaque action est enregistrée avec l'identifiant de l'utilisateur, l'entité concernée, une description, la date et une empreinte de l'adresse IP. Cette combinaison associe contrôle préventif et preuve a posteriori.

CONTRÔLES DE SÉCURITÉ APPLICATIVE

Authentification et session

Les mots de passe doivent compter au moins douze caractères et contenir une majuscule, une minuscule, un chiffre et un symbole. Le stockage utilise Argon2id lorsque l'algorithme est disponible dans PHP, puis PASSWORD_DEFAULT en solution de repli. L'OWASP recommande Argon2id pour les nouvelles applications et rappelle que les mots de passe ne doivent pas être stockés en clair [4]. La fonction password_hash de PHP produit un condensat à sens unique et prend en charge Argon2id lorsque PHP a été compilé avec ce support [5].

Après huit échecs provenant de la même empreinte IP sur quinze minutes, une nouvelle tentative est refusée. Une connexion réussie régénère l'identifiant de session. Le cookie est marqué HttpOnly et SameSite Strict ; l'attribut Secure est activé lorsque l'application fonctionne en HTTPS. Ces choix réduisent les risques de fixation de session, d'accès au cookie par un script et d'envoi involontaire dans un contexte intersite.

Entrées sorties et base de données

Les opérations SQL passent par des requêtes préparées PDO, avec paramètres séparés du texte SQL. Les valeurs affichées dans les vues sont échappées avec htmlspecialchars en mode ENT_QUOTES et encodage UTF 8. Les formulaires qui modifient l'état de l'application contiennent un jeton aléatoire conservé en session et comparé par hash_equals côté serveur. Cette mise en œuvre suit le principe du synchronizer token recommandé pour les applications avec session [6].

En-têtes de sécurité

L'application émet une politique de sécurité du contenu qui limite les scripts, styles, images, cadres, médias et workers à des origines définies. Elle ajoute X Content Type Options, X Frame Options, Referrer Policy et Permissions Policy. Lorsque HTTPS est détecté, Strict Transport Security est envoyé pour les requêtes futures. Ces en-têtes réduisent la surface d'exploitation du navigateur, mais leur efficacité dépend d'une configuration HTTPS correcte et d'une politique régulièrement vérifiée [2].

Téléversement et stockage

Les fichiers publics sont acceptés selon une liste de types MIME détectés par le serveur. Les images sont limitées à 10 Mo, les PDF à 25 Mo et les vidéos à 200 Mo. Le nom d'origine n'est pas utilisé comme nom de stockage ; une valeur aléatoire est générée. Les pièces des candidatures sont limitées à 10 Mo et conservées dans un répertoire privé situé hors de l'arborescence publique. Ces mesures correspondent aux recommandations essentielles de l'OWASP sur la validation, le renommage, les limites de taille et le stockage des fichiers [7]. Elles doivent être complétées par une analyse antivirus ou un mécanisme de désarmement des documents lorsque l'infrastructure le permet.

PUBLICATION SCIENTIFIQUE ET LECTURE CONTRÔLÉE

Le modèle de données impose une relation entre l'article et journal_issues. L'interface d'administration exige donc la création d'une revue, d'un volume ou numéro, puis la sélection de ce numéro avant l'enregistrement de l'article. Le DOI reste facultatif, car tous les articles institutionnels n'en possèdent pas. Cette structure évite de publier une liste d'articles sans contexte éditorial et facilite le classement par revue, volume, numéro, ISSN et date.

L'autorisation de téléchargement est portée par le champ allow_pdf_download. Lorsque sa valeur est active, un point d'accès dédié vérifie que le contenu est un article publié, enregistre l'événement de téléchargement et renvoie le fichier comme pièce jointe. Aucun compte public ni demande préalable n'est exigé. Lorsque l'option est inactive, le lecteur natif du navigateur est remplacé par PDF.js, une bibliothèque qui interprète le document et rend ses pages dans un canvas [11]. L'interface limite la navigation aux quatre premières pages, bloque le menu contextuel ainsi que les raccourcis usuels d'impression et d'enregistrement, et remplace le contenu imprimé par un avertissement. Le flux de lecture utilise une adresse signée et temporaire, refuse la mise en cache et reste limité à la même origine.

Cette protection doit être qualifiée avec précision. Pour afficher une page, le navigateur reçoit nécessairement des données. Un utilisateur avancé peut inspecter les échanges réseau ou capturer l'écran. Le mécanisme livré

réduit les copies accidentelles et supprime les fonctions visibles du lecteur natif, mais il ne constitue pas une gestion numérique des droits. Une protection plus forte demanderait la génération côté serveur d'un fichier d'aperçu ne contenant que les quatre premières pages, ou la production d'images filigranées distinctes du document original.

STATISTIQUES AUDIT ET CONTINUITÉ

Les consultations publiques alimentent la table page_views. L'adresse IP et l'agent utilisateur ne sont pas conservés en clair : une empreinte dérivée avec la clé de l'application est enregistrée. Le tableau de bord calcule les visiteurs uniques, les consultations, la moyenne de pages par visiteur et les pages les plus consultées sur une période de trente jours. Les téléchargements sont enregistrés séparément. Cette distinction permet d'évaluer la portée des contenus sans confondre lecture et récupération d'un document.

Le journal d'audit conserve les actions importantes de l'espace d'administration. Les notifications informent le Owner et le Super Admin lorsqu'un contenu est créé, modifié, soumis ou publié. Cette journalisation soutient l'analyse d'incident et le contrôle du processus éditorial, conformément au rôle attribué aux journaux applicatifs par l'OWASP [8]. Les données consignées doivent néanmoins faire l'objet d'une politique de conservation et d'un contrôle d'accès, car un journal peut lui-même contenir des informations sensibles.

Le service de sauvegarde extrait la structure et les données de MySQL, puis ajoute les médias publics et les pièces privées dans une archive ZIP ou TAR GZ selon les extensions disponibles. MySQL distingue les sauvegardes logiques des sauvegardes physiques et recommande une politique régulière assortie de procédures de récupération [10]. Une archive créée avec succès n'est donc qu'une première étape. Sa copie hors serveur et un exercice de restauration sont nécessaires pour établir qu'elle peut réellement soutenir la reprise.

RÉSULTATS DE LA VÉRIFICATION

La vérification statique montre que les exigences principales disposent d'un mécanisme identifiable dans le code ou le schéma de données. La couverture est plus solide pour les autorisations, les états éditoriaux, la protection CSRF, les requêtes préparées, la validation des fichiers et la journalisation. Elle reste partielle pour les domaines qui exigent une procédure d'exploitation ou une validation externe, notamment l'analyse antivirus, la restauration des sauvegardes, l'accessibilité complète et la résistance à une attaque active.

Tableau 3 — Matrice synthétique de vérification

Domaine	Mécanisme observé	Appréciation	Condition restante
Autorisation	Quatre rôles et vérification par	Implémenté	Revue périodique des comptes

	opération		
Cycle éditorial	Brouillon, soumission, publication et retrait	Implémenté	Procédure interne de validation
Base de données	PDO préparé et émulation désactivée	Implémenté	Tests ciblés d'injection
Formulaires	Jeton CSRF de session et comparaison constante	Implémenté	Tests sur chaque action sensible
Téléversements	Liste MIME, limites, noms aléatoires, stockage privé	Partiel	Antivirus ou CDR absent
Documents scientifiques	Numéro obligatoire, DOI, téléchargement contrôlé	Implémenté	Aperçu serveur recommandé
Traçabilité	Audit, notifications, statistiques et empreintes	Implémenté	Politique de conservation
Continuité	Archive SQL, médias et dossiers privés	Partiel	Copie hors site et restauration testée
Accessibilité	Structure responsive et libellés principaux	Partiel	Audit WCAG 2.2 nécessaire

Le portail couvre également les besoins de contenu qui entourent la sécurité : gestion bilingue français anglais, diaporama administrable, filières et orientations, valve hebdomadaire, comité de gestion, secteurs et bureaux, médias, revues, chercheurs, inscriptions et réseaux sociaux. Cette intégration évite de multiplier les applications et les identifiants. Elle concentre cependant le risque sur un même système, ce qui rend indispensables la mise à jour de PHP, des bibliothèques JavaScript et du serveur Apache.

DISCUSSION

Apports de la conception

Le premier apport est la traduction d'une organisation réelle en règles applicatives. Les rôles ne sont pas de simples étiquettes affichées dans l'interface : ils conditionnent la création, l'édition, la publication, la

suppression, la gestion des comptes et l'accès aux paramètres. Le deuxième apport concerne la publication scientifique. La revue et son numéro deviennent des objets gérés avant les articles, ce qui rend l'organisation éditoriale explicite dans la base. Le troisième apport est la traçabilité commune aux contenus, téléchargements, consultations et candidatures.

Le choix de PHP, MySQL, JavaScript, CSS et HTML répond à une contrainte d'exploitation. Ces technologies sont disponibles dans XAMPP et peuvent être administrées localement sans infrastructure complexe. Cette simplicité facilite le déploiement, mais elle ne dispense pas de séparer l'environnement de développement du serveur de production, de protéger les secrets et de désactiver les informations de débogage en exploitation.

Limites

Quatre limites empêchent de considérer l'évaluation comme une certification de sécurité. Premièrement, l'inspection du code n'a pas été complétée par un test d'intrusion indépendant. Deuxièmement, les statistiques de fréquentation n'ont pas encore servi à une étude longitudinale et ne permettent pas d'estimer l'adoption. Troisièmement, l'interface n'a pas fait l'objet d'un audit formel selon WCAG 2.2, que le W3C recommande comme version actuelle des règles d'accessibilité [9]. Quatrièmement, la lecture PDF protégée agit dans le navigateur et ne peut empêcher une capture ou une récupération avancée du flux.

La sauvegarde présente une limite similaire entre fonction et assurance. Le système sait produire une archive, mais seule une restauration réussie sur un environnement distinct prouvera la capacité de reprise. Enfin, le contrôle des fichiers reconnaît leur type et leur taille sans analyser leur contenu malveillant. Ces limites ne rendent pas l'artefact inutilisable ; elles définissent les travaux nécessaires avant de lui confier des données plus sensibles ou un volume plus important.

Recommandations pour le déploiement

1. Forcer HTTPS sur tout le domaine, vérifier l'en-tête HSTS et renouveler automatiquement le certificat.

2. Ajouter une authentification multifacteur pour le Owner et le Super Admin, puis revoir les comptes actifs à intervalles définis.

3. Générer côté serveur un document d'aperçu limité aux quatre premières pages afin que le fichier complet ne soit jamais transmis lorsque son téléchargement est interdit.

4. Soumettre les fichiers téléversés à un antivirus ou à un service de désarmement, en particulier les PDF des candidatures et les documents scientifiques.

5. Copier les sauvegardes sur un support distinct, chiffrer les archives sensibles et exécuter périodiquement une

restauration complète.

6. Mettre en place des tests automatisés sur les autorisations, les jetons CSRF, les téléversements, les transitions de statut et les routes de téléchargement.
7. Réaliser un audit d'accessibilité WCAG 2.2 et des essais avec des étudiants, des agents et des administrateurs sur téléphone et ordinateur.
8. Définir la durée de conservation des candidatures, journaux, statistiques et sauvegardes, ainsi que les personnes habilitées à les consulter.

Conclusion

Le portail de l'ISTM Nyankunde montre qu'une application PHP et MySQL de taille maîtrisée peut intégrer la communication institutionnelle, l'enseignement, la recherche et l'admission dans un même environnement administrable. La séparation des rôles, le circuit de validation, les requêtes préparées, les jetons CSRF, le contrôle des fichiers, les en-têtes de sécurité, les journaux et les sauvegardes apportent une amélioration concrète par rapport à une publication web sans gouvernance explicite.

La principale conclusion concerne la relation entre sécurité et organisation. Les contrôles techniques sont efficaces lorsqu'ils traduisent une responsabilité institutionnelle claire : qui crée, qui valide, qui publie, qui consulte et qui répond d'une action. La prochaine étape ne consiste pas à ajouter des fonctions visibles, mais à démontrer la qualité opérationnelle du système par des tests d'intrusion, des restaurations, un audit d'accessibilité et une observation de l'usage réel. Ces vérifications permettront de transformer l'artefact fonctionnel en service institutionnel durable.

Références

[1] Peffers K, Tuunanen T, Rothenberger M A et Chatterjee S. A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), 45 à 77, 2007. https://doi.org/10.2753/MIS0742-1222240302

[2] OWASP Foundation. Application Security Verification Standard, version 5.0.0. 2025. https://owasp.org/www-project-application-security-verification-standard/

[3] National Institute of Standards and Technology. Role Based Access Control. Présentation du modèle et du standard INCITS 359. https://csrc.nist.gov/Projects/role-based-access-control/faqs

[4] OWASP Foundation. Password Storage Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

[5] PHP Documentation Group. password_hash. Manuel PHP. https://www.php.net/manual/en/function.password-hash.php

[6] OWASP Foundation. Cross Site Request Forgery Prevention Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

[7] OWASP Foundation. File Upload Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html

[8] OWASP Foundation. Logging Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html

[9] World Wide Web Consortium. Web Content Accessibility Guidelines WCAG 2.2. Recommandation du 5 octobre 2023. https://www.w3.org/TR/WCAG22/

[10] Oracle. MySQL 8.4 Reference Manual. Backup and Recovery. https://dev.mysql.com/doc/refman/8.4/en/backup-and-recovery.html

[11] Mozilla Foundation. PDF.js Getting Started. https://mozilla.github.io/pdf.js/getting_started/

Sources en ligne consultées le 7 septembre 2026.