DOI 10.15276/imms.v13.no3-4.349
УДК 004.056.53

Informatics and Mathematical Methods in Simulation
Vol.13 (2023), No. 3-4, pp. 349-359

РОЗРОБКА ЗАХИЩЕНОЇ СИСТЕМИ ДЛЯ ОБМІНУ ДОКУМЕНТАМИ У НАВЧАЛЬНОМУ ПРОЦЕСІ

І.М. Чураков, Н.І. Кушніренко, В.В. Зоріло

Національний університет «Одеська політехніка»
1 Шевченка пр., Одеса, 65044, Україна
e-mail: infsec2011@gmail.com

Головним акцентом дослідження є потреба в сучасній та ефективній системі обміну документами між викладачами та студентами. Зміни в освітньому оточенні, структурі та форматі навчального процесу в результаті поширення технологій і зокрема умов пандемії COVID-19 вимагають новітніх рішень. Ефективність навчального процесу та його керованість значною мірою залежать від швидкості, надійності та зручності обміну інформацією між учасниками освітнього процесу. В освітніх закладах щодня циркулюють величезні обсяги інформації від пересилки лабораторних робіт до контрольно-підсумкових завдань. Без сучасних цифрових технологій, цей процес може бути не тільки повільним, але і супроводжуватись втратами, помилками і затримками, тому необхідність системи обміну документами є актуальною на сьогодні. Крім того, великою проблемою для впровадження таких систем є питання інформаційної безпеки, адже обмін документами включає обробку та передачу конфіденційної інформації, яка повинна бути захищена від несанкціонованого доступу та зловмисного використання. У зв'язку з цими викликами, було зосереджено зусилля на розробці системи обміну документами, яка використовує сучасні алгоритми шифрування для безпечної передачі даних. Це дослідження та розробка набули не тільки академічної, а й практичної значимості, оскільки вони заповнюють прогалину між вимогами до сучасного електронного навчання і реальними можливостями освітніх установ в обміні та захисті інформації. Результатом проведеного дослідження стала розробка системи, яка використовує сучасні цифрові технології, такі як двофакторна авторизація, протокол шифрування AES та алгоритм хешування PBKDF2. Розроблена система обміну документами – це практичний внесок в галузь освітніх технологій, який відкриває нові горизонти для автоматизації, ефективності та безпеки в сфері обміну документами в освіті.

Ключові слова: двофакторна авторизація, API-ключі, PBKDF2, безпека даних, управління базою даних, обмін документами, електронне навчання.

Вступ. Ми живемо в епоху електронних технологій та глобалізації, коли цифрові інструменти стають нероздільною частиною як професійної діяльності, так і повсякденного життя [1]. Сучасне освітнє середовище теж не залишається за межами цього тренду. Навчальні заклади в усьому світі активно інтегрують технології в навчальний процес, намагаючись зробити його більш ефективним і доступним.

Одним з найважливіших напрямків цифрового впровадження в освіті є розробка систем обміну документами [2]. Це є необхідною складовою успішного навчального процесу, адже передача документації між студентами, викладачами та адміністрацією стає швидкою та ефективною. Але з ростом кількості персональних та конфіденційних даних, які пересилаються через цифрові канали, виникає проблема забезпечення інформаційної безпеки.

Не можна ігнорувати ризики, пов'язані з потенційними кібератаками чи несанкціонованим доступом до інформації. Тому важливо розуміти, що цифрова

трансформація освіти вимагає не тільки впровадження новітніх технологій, але і розробки ефективних механізмів їх захисту.

Обмін документами в освітній сфері став основою багатьох досліджень. Наприклад, багато таких інструментів як Google Classroom, Moodle та Remind вже успішно використовуються в навчальних закладах. Google Classroom, один з найпопулярніших інструментів, дозволяє викладачам й учням надсилати та отримувати завдання, проводити тести та швидко обмінюватися повідомленнями [3]. Moodle, відкрита система керування курсами, пропонує більш налаштовувану систему з великою кількістю модулів та плагінів, включаючи безпосередній обмін файлами [4]. Remind, сервіс для надсилання повідомлень, дозволяє викладачам й учням миттєво обмінюватися інформацією без розкриття особистих контактних даних [5]. Однак, хоча ці системи вже розширюють можливості обміну документами в освіті, вони можуть бути перевантажені зайвим функціоналом і часто є платними. В той же час більш прості програми не вирішують в повній мірі питання інформаційної безпеки. В даній роботі запропонована система обміну документами, яка забезпечує захист інформації, що в ній циркулює.

Мета статті та постановка завдань. В сучасному світі інформаційних технологій і цифрової глобалізації, область освіти стикається з новими викликами, зокрема, із необхідністю безпечного обміну документами в процесі навчання. Це потребує ретельного вивчення та аналізу, якому ми присвячуємо нашу роботу.

Метою роботи є аналіз та вибір найбільш ефективних засобів для забезпечення безпеки даних в процесі обміну документами, а також розробка та впровадження обраних засобів в систему обміну документами.

Для досягнення поставленої мети необхідно розв’язати наступні *задачі*:

1) Проаналізувати засоби для забезпечення безпеки даних.
2) Оцінити можливості різних систем керування реляційними базами даних.
3) Вибрати спосіб реалізації системи авторизації.
4) Впровадити обрані підходи в розроблювану систему для обміну документами.

Особливий акцент в нашій роботі буде зроблено на питання захисту персональних даних учасників освітнього процесу, оскільки це критично для дотримання принципів конфіденційності та захисту прав людини.

Основна частина. Забезпечення безпеки даних – це один з ключових елементів розробки програмного забезпечення. Це включає в себе шифрування даних, забезпечення цілісності даних, та системи аутентифікації та авторизації.

Серед доступних варіантів наступні вирізняються своїми перевагами та недоліками:

1. Шифрування даних:
    1.1. AES (Advanced Encryption Standard): Стандарт шифрування, використовується урядами, банками та іншими організаціями, яким потрібна висока ступінь захисту даних. Проте, він є складним у використанні і потребує певних навичок для безпечного застосування.
    1.2. RSA (Rivest - Shamir - Adleman): Це широко використовується варіант асиметричного шифрування, що дозволяє безпечне обмін даними. Його недоліком є те, що він повільніший за AES.
2. Хешування даних:
    2.1. PBKDF2 (Password-Based Key Derivation Function 2) дозволяє створити криптографічно безпечний хеш від вхідного пароля. Недоліком є те, що його можна «зламати» з достатньо потужним обчислювальним обладнанням[6].
    2.2. bcrypt: Це ще одна популярна функція хешування для захисту паролів. Вона вважається безпечнішою, але є повільнішою за PBKDF2.

3. Аутентифікація:

3.1. Двофакторна аутентифікація: Це потужний спосіб забезпечити безпеку облікового запису, вимагаючи додаткового кроку підтвердження після введення пароля. Але він також може збільшити складність використання системи для кінцевого користувача.

3.2. Безпечний обмін ключами (наприклад протокол Діффі-Хеллмана). Це дозволяє двом сторонам безпечно обмінюватися ключами по надійному каналу. Недолік – це складність реалізації.

Вибір конкретних методів завжди надає багато простору для обговорення, але для цього проекту вирішено було скористатись шифруванням даних AES, хешуванням PBKDF2 та двофакторною автентифікацією.

AES вважається одним з найбільш надійних алгоритмів шифрування, використовуючи блочне шифрування з ключем довжиною від 128 до 256 біт для надійного захисту даних [7]. Незважаючи на те, що це може бути складніше для реалізації, використання AES забезпечує високий рівень захисту даних.

PBKDF2, у свою чергу, є стандартом для хешування паролів і використовується дуже широко. Його основна перевага полягає в використанні солі для захисту від швидких атак на хеші паролів. Це означає, що навіть якщо зловмисник отримає хеші паролів, йому все одно доведеться витратити значний час і ресурси для злому кожного окремого пароля.

Двофакторна автентифікація, хоча і може здатися складною для користувачів, забезпечує додатковий шар захисту, зменшуючи шанс несанкціонованого доступу до системи навіть у випадку витоку або викрадення пароля.

Таким чином, комбінація обраних нами трьох методів створює ідеальний баланс між надійністю та зручністю користування саме для задач даного проекту.

У контексті обраної нами теми, доцільно розглянути інструмент, який буде гарантувати безпеку даних – систему керування реляційними базами даних (RDBMS).

Система керування реляційними базами даних (RDBMS) – це програмне забезпечення, що дозволяє створювати, оновлювати та керувати реляційною базою даних. На сьогоднішній день є велика кількість RDBMS, серед найпопулярніших: MySQL, PostgreSQL, SQLite, Microsoft SQL Server.

Важливо обрати RDBMS, яка найбільше підходить для проекту, враховуючи такі чинники, як розмір проекту, очікуване навантаження на базу даних, вимоги до професійних навичок розробників, вартість, та багато інших [8]. В табл. 1 приведено порівняльну таблицю деяких з найбільш популярних реляційних СКБД [9], що використовуються сьогодні, з окресленням їхніх основних переваг та недоліків.

Таблиця 1

Порівняльна характеристика систем керування реляційними базами даних

RDBMS	Переваги	Недоліки
MySQL	Універсальність, швидкість, висока продуктивність і стійкість.	Є певні обмеження у функціональності.
PostgreSQL	Гнучкість, велика кількість можливостей, відкритий код.	Складніший в налаштуванні, вимагає більше ресурсів машини.

продовження таблиці 1

SQLite	Невеликий розмір, проста в використанні, не вимагає окремого сервера.	Обмежена місткість, не підходить для великих проектів.
Microsoft SQL Server	Тісна інтеграція з .NET, відмінна підтримка, надійність.	Платна, може бути занадто громіздкою для малих проектів.

Після детального вивчення наведених систем керування реляційними базами даних було вирішено скористатись MySQL для цього проекту.

MySQL є універсальним рішенням, відомим своєю швидкістю, надійністю та стійкістю. Її використовують десятки тисяч веб-сайтів по всьому світу, включаючи таких великих гравців як Facebook, Twitter і YouTube.

Незважаючи на певні обмеження у функціональності, що можуть бути знайдені в інших RDBMS, MySQL надає усі необхідні можливості для ефективного керування даними в рамках цього проекту. Вона має зрозумілий інтерфейс і легко інтегрується з іншими технологіями, що використовуються на проекті.

Вибір MySQL і для цього проекту також обумовлений розглядом таких чинників як розмір та складність проекту, плановане навантаження на базу даних. Однак, організація безпечного зберігання даних - це лише один аспект багатогранного процесу створення надійного і складного проекту. Окрім цього, ефективність системи в значній мірі залежить від правильно вибраної системи авторизації.

Система авторизації є ключовою складовою безпеки будь-якої сучасної системи або додатку. Вона перш за все повинна забезпечити впевненість, що тільки відповідні користувачі мають доступ до відповідних ресурсів. Різні системи мають різні вимоги до авторизації, тому важливо обрати правильний метод, що найкраще відповідає потребам.

Існують такі основні методи реалізації системи авторизації:

• JWT (JSON Web Tokens): цей метод передбачає використання токенів для авторизації користувачів. Безпека JWT провіряється за допомогою підпису токена. Головною перевагою є те, що сервер не потребує зберігати сесію користувача, тобто це без станова авторизація [10].

• Сесії: цей метод передбачає створення унікального ідентифікатора сесії, який зберігається на сервері і використовується для перевірки привілеїв користувача. Цей підхід вимагає більше ресурсів, але надає більший контроль над даними сесії.

• OAuth: це стандарт, що дозволяє користувачам надавати застосункам обмежений доступ до своїх ресурсів на інших сайтах. Це складніший, але більш гнучкий метод авторизації, який широко використовується для розподілених систем [11].

• OTP (One-Time Password): цей метод передбачає використання одноразового пароля, який генерується алгоритмом і має короткий термін дії. OTP надійний, оскільки динамічні паролі важче викрасти, а також не можна використовувати повторно.

Реалізація системи авторизації на основі одноразових паролів (OTP) для цього проекту була обрана з кількох причин.

По-перше, OTP надає високий рівень безпеки в порівнянні зі сталими паролями. Кожний пароль використовується тільки один раз, тому навіть якщо зловмисник перехопить пароль, він не зможе повторно використати його. Це важливо в контексті глобального зростання кіберзлочинності.

По-друге, використання OTP доповнює інші заходи безпеки, що були вибрані для цього проекту. Комбінація OTP і двофакторної аутентифікації створює сильний захист від несанкціонованого доступу.

По-третє, більшість людей вже знайомі з концептом OTP через їх використання в банківському секторі і інших сферах, де вимагається висока безпека. Отже, використання OTP не створить бар'єрів для використання системи кінцевими користувачами.

Ось чому, вимірюючи переваги і недоліки різних систем авторизації, ми вирішили, що OTP наразі найкраще підходить для цього проекту. Але не можна забувати, що ефективна система авторизації повинна також передбачати додаткові рівні захисту, як, наприклад, двофакторна авторизація.

Специфіка 2FA полягає в тому, що для успішної авторизації потрібно надати не тільки щось, що вам відомо (наприклад, пароль), але і щось, що ви маєте (наприклад, мобільний телефон). Це значно підвищує безпеку системи, тому що потенційний зловмисник має не тільки вичислити пароль, але і фізично отримати пристрій користувача [12].

Для цього проекту було обрано комбінований підхід до 2FA, який використовує одноразові паролі (OTP) та програму Google Authenticator.

Принцип роботи Google Authenticator та одноразового пароля [13]:

1. Генерація Ключа. Коли користувач реєструє обліковий запис на підтримуваному веб-сервісі, генерується унікальний секретний ключ. Цей ключ зберігається на сервері веб-служби, а також передається на пристрій користувача, часто у вигляді QR-коду.

2. Сканування QR-коду. Для початкового налаштування Google Authenticator, користувачу необхідно сканувати QR-код своєю мобільною камерою. Цей QR-код містить таємний ключ.

3. Генерування OTP. За допомогою секретного ключа, Google Authenticator генерує шестизначний код OTP. Цей код змінюється кожні 30 секунд. Google Authenticator використовує стандартний алгоритм HMAC-SHA1 для генерації OTP.

4. Введення OTP. Коли користувач намагається увійти в свій обліковий запис, він вводить OTP, показаний в Google Authenticator, разом із своєю поштою та паролем.

5. Перевірка OTP: Після того, як користувач вводить OTP, сервер, з якого було отримано ключ, використовує той же алгоритм, що і Google Authenticator на базі ключа та поточного часу, щоб перевірити вхідний OTP.

6. Автентифікація При успішному введенні OTP: OTP приймається і користувач авторизується в сервіс.

Тим самим, використання Google Authenticator та одноразових паролів для досягнення двофакторної авторизації в системі забезпечує посилений рівень захисту. За такої схеми, навіть якщо зловмисник вдасться отримати основний пароль користувача, він все одно не зможе здійснити несанкціонований вхід, не маючи доступу до конкретного пристрою, на який надіслано OTP. Але важливо згадати, що для забезпечення загальної безпеки системи, необхідно також акцентувати увагу на безпеці інших елементів, таких як, наприклад, API ключі.

Отримання та використання API-ключів є необхідною складовою більшості веб-систем. Однак, неправильне зберігання чи передача API ключів може призвести до серйозних проблем з безпекою, таких як несанкціонований доступ до системи чи витік даних. Тому забезпечення безпеки API ключів є критично важливим.

Одним із способів безпечного зберігання API ключів є їх шифрування перед зберіганням за допомогою надійних шифрувальних алгоритмів. В даному проекті для цього використовується алгоритм AES (Advanced Encryption Standard) [14].

AES – це симетричний алгоритм блочного шифрування, що використовує однакові ключі для шифрування та дешифрування даних. Базова концепція цього алгоритму полягає в заміні, перестановці та перетворенні інформації з метою забезпечення конфіденційності [15].

Для зберігання API ключів використовується такий процес:

- Генерація API ключа. Сервер генерує унікальний API ключ для кожного користувача або сесії.
- Шифрування API ключа. API ключ шифрується за допомогою алгоритму AES за допомогою вибраного ключа шифрування.
- Зберігання шифрованого API-ключа. Шифрований API-ключ зберігається в захищеному місці, такому як база даних або середовище.
- Дешифрування API ключа при використанні. Коли API ключ потрібно використати, він дешифрується за допомогою того ж шифрувального ключа, що й при шифруванні.

За допомогою цього підходу, API ключі зберігаються на сервері в безпечному вигляді і можуть бути безпечно передані через комунікаційні канали, забезпечуючи конфіденційність і цілісність даних.

Хоча AES є потужним алгоритмом шифрування, використовуваним у багатьох сучасних системах, існують інші алгоритми шифрування, які також можуть бути застосовані в залежності від специфічних вимог до безпеки.

В кожному випадку при виборі алгоритму шифрування важливо врахувати ряд параметрів, таких як тип шифрування (симетричний або асиметричний), розмір ключа та розмір блоку. Розмір ключа є важливим аспектом, який визначає рівень безпеки шифрування - більший розмір ключа збільшує кількість можливих ключів і мінімізує ризик злому шляхом «прямого пошуку».

Як приклад, в таблиці 2 наведено ряд алгоритмів шифрування та їх ключові характеристики:

Таблиця 2

Порівняння основних алгоритмів шифрування

Алгоритм	Тип	Розмір ключа	Розмір блоку
AES	Симетричний	128, 192 або 256 біт	128 біт
DES	Симетричний	56 біт	64 біт
Triple DES	Симетричний	112 або 168 біт	64 біт
RSA	Асиметричний	1024 або 2048 біт	Змінний по блокам
ElGamal	Асиметричний	1024 або 2048 біт	Змінний по блокам
Blowfish	Симетричний	32-448 біт	64 біт

Використовуючи зазначені вище технології, було розроблено програму на платформі Windows Forms спеціально для потреб викладачів. Ця програма спрямована на значне поліпшення процесу обміну документами між викладачами та студентами, пропонуючи цілий ряд функціональних можливостей.

Центральними елементами програми є системи реєстрації та аутентифікації користувачів. Алгоритм зашифрованої системи реєстрації обробляє всю необхідну інформацію та зберігає її в базі даних MySQL, використовуючи для цього стандарт AES. Тимчасом, двофакторна система аутентифікації, що ґрунтується на One-Time Password (OTP) та Google Authenticator, забезпечує високий рівень безпеки користувачів' даних.

Програмний продукт, який було розроблено, є сучасною та ефективною системою для управління студентською базою даних та спілкування зі студентами.

У програмі реалізовані логін та реєстрація для забезпечення зручності та безпеки користувачів. При цьому, програма представляє три ключові функціонали.

Додавання студентів в базу даних. В даному розділі програми користувачеві надається можливість вносити інформацію про студентів, а також прив'язувати їх до відповідних груп і предметів. Це забезпечує легкість управління та зручне пошук інформації про кожного студента. Форму з реалізацією даного функціоналу зображено на рисунку 1.

Відправка електронних листів студентам. В цьому розділі програми користувачеві надається можливість автоматично відправляти листи студентам з прикріпленими файлами, а також відправляти завдання у випадковому порядку. Це полегшує процес взаємодії зі студентами і забезпечує їх оперативне інформування. Форму з реалізацією даного функціоналу зображено на рисунку 2.

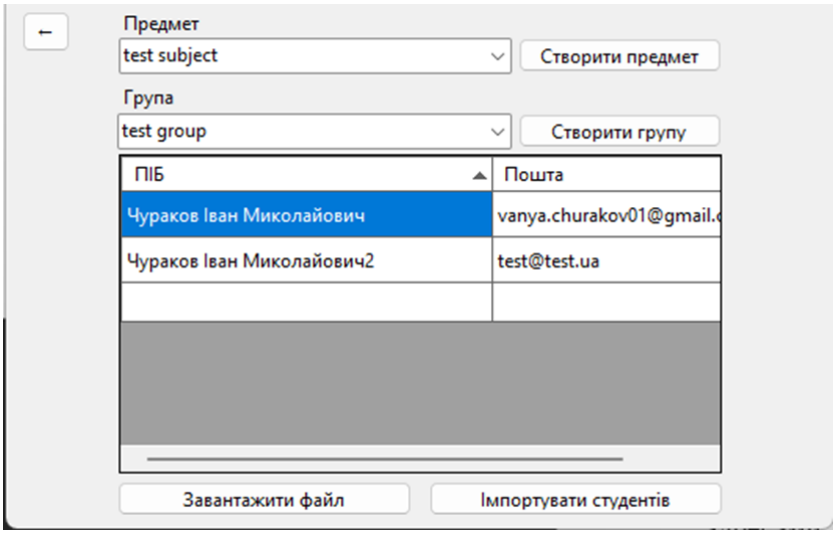

Рис. 1. Форма імпорту студентів

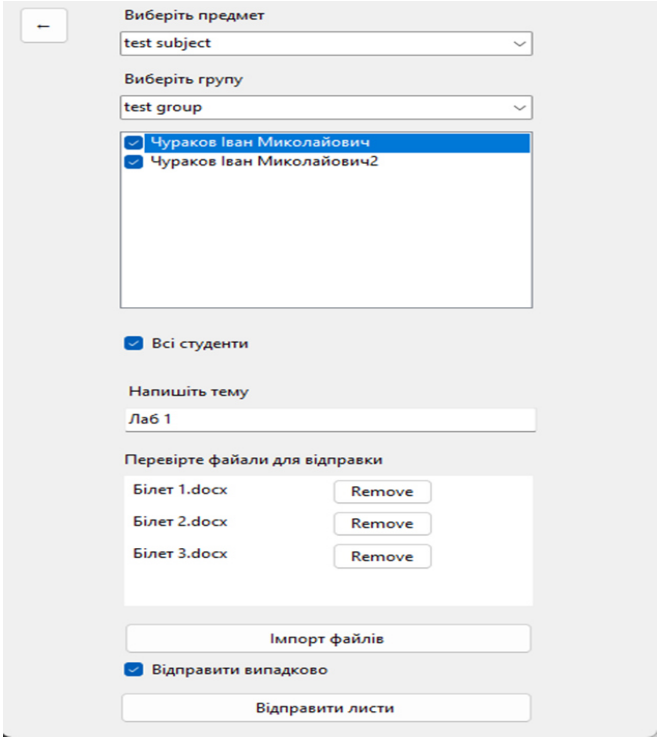

Рис. 2. Форма відправки листів

Перегляд відправлених завдань. У цьому розділі програма надає можливість переглядати, кому були відправлені які завдання та які студенти вже здали свої роботи. Це дозволяє контролювати процес виконання завдань та забезпечує зворотний зв'язок. Форму з реалізацією даного функціоналу зображено на рисунку 3.

Так при обмежених ресурсах, але з врахуванням сучасних технологій ми змогли розробити ефективний та надійний продукт.

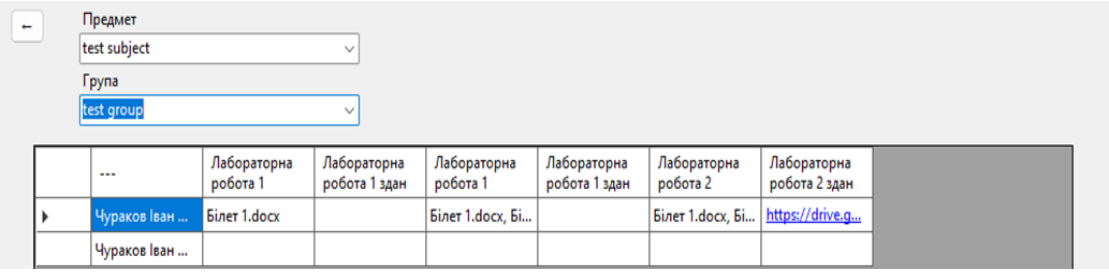

Рис. 3. Форма дошки статусів

Висновки. В роботі проведена розробка системи для захищеного обміну документами між викладачем та студентами.

Для досягнення поставленої мети було розв’язано наступні задачі:

1. Виконано аналіз сучасних рішень для автоматизації документообігу в навчальному процесі. Виявлено, що існуючі програмні рішення зазвичай перенавантажені зайвим функціоналом, а більш прості програми не завжди у повній мірі забезпечують захист конфіденційних даних користувачів.
2. Виконано аналіз сучасних технологій, які застосовуються для розробки захищеного програмного забезпечення. Для розроблюваної системи обрано багатофакторну аутентифікацію за допомогою Google Authenticator для підвищення ступеня захисту системи, що реалізує додатковий рівень перевірки автентифікації користувача. Було проаналізовано та застосовано алгоритм шифрування AES для зберігання API-ключів з метою безпечного зберігання цих ключових ідентифікаторів, що мінімізують ризик непередбаченого витоку конфіденційних даних. Впроваджено алгоритм PBKDF2 для хешування паролів.
3. Розроблена система має наступні можливості:
   - реєстрація користувачів;
   - імпорт та редагування даних студентів та груп;
   - відправка електронних листів студентам;
   - відправка файлів в різних форматах. підтримка відправки в звичайному і випадковому режимах; можливість зберігання документів викладачів в хмарному сховищі;
   - можливість завантаження студентами результатів робіт у вигляді файлів в виділену папку в хмарному сховищі;
   - перегляд відправлень та статусів виконання завдань на дошці статусів.

Як підсумок, можемо зазначити, що запропонована система автоматизує рутинний процес обміну документами між викладачами та студентами і тим самим дозволяє зекономити час та мінімізувати ймовірність помилки, при цьому забезпечується захист конфіденційної інформації користувачів. Розроблений програмний продукт має зручний та інтуїтивно зрозумілий інтерфейс. В якості елементів системи було використано сучасне відкрите програмне забезпечення. Дана система є гнучкою та легко масштабується, що дозволяє шляхом невеликих змін адаптувати її до інших сфер.

Список літератури

1. Vincent J. Life stage or Age. Reviewing perceptions of oldest digital technologies users. *Digital Ageism*. 2023. P. 36-52. DOI:10.4324/9781003323686-3.
2. Bejinaru R. Impact of digitalization on education in the knowledge economy. *Management Dynamics in the Knowledge Economy*. 2019. V.7.3. P. 367-380.
3. About Classroom. URL: https://support.google.com/edu/classroom/answer/6020279?hl=en
4. Features. URL: https://docs.moodle.org/403/en/Features
5. A New Chapter Begins. URL: https://www.remind.com/parentsquare
6. Ertaul L., Manpreet K., Venkata A.K.R. Gudise. Implementation and performance analysis of pbkdf2, bcrypt, scrypt algorithms. *Proceedings of the international conference on wireless networks (ICWN)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016. http://borg.csueastbay.edu/~lertaul/PBKDFBCRYPTCAM READYICWN16.pdf
7. Singh A. Comparative study of DES, 3DES, AES and RSA. *Int. J. Comput. Technol*. 2013. V.9.3. P. 97-102.
8. Hong S. Big Data: how to choose the right cloud infrastructure. 2016. URL: https://www.researchgate.net/publication/305495731_Big_Data_how_to_choose_the_right_cloud_infrastructure
9. Khawar I. Huge and Real-Time Database Systems: A Comparative Study and Review for SQL Server 2016, Oracle 12c & MySQL 5.7 for Personal Computer. *Journal of Basic & Applied Sciences*. 2017. No.13. P. 481-490.
10. Rajat S. Understanding JSON Web Token Authentication. URL: https://medium.com/p/a1febf0e15
11. Polu S. K. OAuth based Secured authentication mechanism for IoT Applications. *International Journal of Engineering Development and Research (IJEDR)*. 2018. V.6. No.4. P. 409-414.
12. Reese K., Smith T., Dutson J., Armknecht J., Cameron J., Seamons K. A usability study of five {two-factor} authentication methods. *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 2019. URL: https://www.usenix.org/system/files/soups2019-reese.pdf
13. Tilak L. Google Authenticator and how it works? URL: https://medium.com/@tilaklodha/google-authenticator-and-how-it-works-2933a4ece8c2
14. Chappel S. Hiding in Plain Sight: A Guide to Keeping Your API Keys Safe. URL: https://medium.com/@sams-scripts/hiding-in-plain-sight-a-guide-to-keeping-your-api-keys-safe-2fba0373d7a8
15. Rijmen V., Daemen J. Advanced encryption standard. *Proceedings of federal information processing standards publications, national institute of standards and technology* 2001. V.19. s.22. URL: https://jima.me/wp-content/uploads/2016/05/Advanced-Encryption-Standard-Wikipedia-the-free-encyclopedia.pdf

І.М. Чураков, Н.І. Кушніренко, В.В. Зоріло

DEVELOPMENT OF A SECURE SYSTEM FOR DOCUMENT EXCHANGE IN THE EDUCATIONAL PROCESS

I. Churakov, N. Kushnirenko, V. Zorilo

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
e-mail: infsec2011@gmail.com

The main focus of the study is the need for a modern and efficient system of document exchange between teachers and students. Changes in the educational environment, structure and format of the educational process as a result of the spread of technology and, in particular, the COVID-19 pandemic require the latest solutions. The effectiveness of the educational process and its manageability largely depend on the speed, reliability and convenience of information exchange between participants in the educational process. Educational institutions exchange huge amounts of information on a daily basis, from moving homework to final exams. Without modern digital technologies, this process can be not only slow, but also accompanied by losses, errors and delays, so the need for a document exchange system is relevant today. In addition, information security is a major anchor for the implementation of such systems, as document exchange involves the processing and transmission of sensitive information that must be protected from unauthorized access and misuse. In response to these challenges, efforts have been focused on developing a document exchange system that uses modern encryption algorithms for secure data transmission. This research and development has acquired not only academic but also practical significance, as it fills the gap between the requirements for modern e-learning and the real capabilities of educational institutions in the exchange and protection of information. The result of the conducted research was the development of a system that uses modern digital technologies, such as two-factor authentication, the AES encryption protocol, and the PBKDF2 hashing algorithm. The developed document exchange system is a practical contribution to the field of educational technologies, which opens new horizons for automation, efficiency and security in the field of document exchange in education.

Keywords: two-factor authentication, API keys, PBKDF2, data security, database management, document exchange, e-learning.