

Юрій ПРОКОПЧУК, студ.
ORCID ID: 0009-0008-7363-2080
e-mail: yurii.prokorchuk@knu.ua
Київський національний університет
імені Тараса Шевченка, Київ, Україна

Олена ФЕНДЬО, канд. техн. наук, доц.
ORCID ID: 0000-0001-9197-6399
e-mail: olena1.fendyo@knu.ua
Київський національний університет
імені Тараса Шевченка, Київ, Україна

ПРОГРАМНА СИСТЕМА НАВЧАННЯ ТА ТЕСТУВАННЯ ЗНАНЬ

У статті розглянуто підходи до створення програмної системи навчання та тестування знань, спрямованої на підвищення ефективності освітнього процесу в умовах цифрової трансформації освіти. Виконано аналіз сучасних теоретичних і методологічних засад е-навчання та тестування. Досліджено особливості функціонування систем управління навчанням (LMS). Розглянуто технологічні рішення, які використовують при їх розробленні та реалізації. Здійснено порівняльний аналіз наявних систем управління навчанням на прикладі Moodle та Canvas. Запропоновано архітектуру та реалізовано вебзастосунок, який забезпечує автоматизацію процесів навчання, контролю та оцінювання знань із використанням сучасних технологій frontend-(React, JavaScript) і backend-розроблення (Java, Spring Boot, PostgreSQL). Створено модуль для рекомендацій курсів на основі уподобань користувача із використанням мови програмування Python та бібліотеки KeyBERT. Одержані результати підтверджують можливість ефективного використання розробленої вебсистеми в закладах освіти та організаціях, які надають освітні послуги.

Ключові слова: е-навчання, е-тестування, система управління навчанням, освітня програмна система, LMS.

Вступ

Сучасна система освіти перебуває у стані активної цифрової трансформації, що зумовлює необхідність використання

інноваційних технологій для забезпечення гнучкості, доступності та якості навчання. Цифрова трансформація освіти передбачає інтеграцію цифрових технологій у всі аспекти викладання, навчання та адміністрування з метою створення більш персоналізованих, ефективних та доступних навчальних програм (Гулай та ін., 2023).

Е-навчання та *е*-тестування стають важливою складовою освітнього процесу, адже дозволяють автоматизувати контроль знань та створювати інтерактивне цифрове середовище для здобувачів освіти.

Одними із головних інструментів реалізації таких підходів є програмні системи управління навчанням (СУН, Learning Management System, LMS), що поєднують функції організації навчального процесу, зберігання матеріалів, проведення тестування. Проте, наявні рішення часто не враховують специфічні потреби закладів освіти або мають обмежені можливості адаптації. Тому актуальним завданням є створення програмної системи, яка забезпечує ефективне навчання та тестування знань із урахуванням сучасних технологічних і педагогічних вимог.

Метою дослідження є підвищення ефективності процесів навчання та тестування знань на основі створеної програмної системи з використанням вебтехнологій. Для досягнення мети необхідно: дослідити теоретичні основи програмних систем навчання та тестування знань; здійснити аналіз програмно-технологічних рішень для їх побудови; розробити, реалізувати та протестувати програмну систему навчання та тестування знань.

Об'єктом дослідження є процеси навчання та тестування знань, а *предметом дослідження* - програмна система навчання та тестування знань.

Методами дослідження є системний аналіз, системний синтез, порівняльний аналіз наявних рішень.

Практичне значення роботи полягає у тому, що створена програмна система дозволяє оптимізувати процес навчання та тестування знань та підвищити гнучкість. Програмна система може бути використана для організації процесів навчання та

тестування знань у закладах освіти або інших організаціях, що надають освітні послуги.

Основна частина і результати

У сучасній системі освіти значного розвитку та поширення набули спрямованість на особистість, залучення учнів до наукового пошуку. Зокрема, активно впроваджуються технології електронного навчання (e-Learning), головним елементом якого є використання інформаційно-комунікаційних технологій. Існують різні способи якими можна реалізовувати навчальний процес. Основними формами є традиційна – заняття проводяться у визначений час у певній аудиторії, дистанційна – учасники освітнього процесу перебувають у різних місцях. Також виокремлюють змішаний формат взаємодій, який передбачає поєднання кожної із вищезгаданих форм (Ребуха та ін., 2022).

Навчання також може відбуватись у синхронному форматі, коли учасники освітнього процесу взаємодіють наживо в реальних або віртуальних аудиторіях, або асинхронному, коли комунікація відбувається із певною затримкою, наприклад, у разі використання пошти чи соціальних мереж (Cleveland-Innes & Wilton, 2018).

Змішана форма навчання дозволяє досягнути значно більшого рівня гнучкості порівняно із традиційними методами організації навчального процесу, оскільки комунікація не обмежується лише часом під час занять. З'являється можливість більш якісно розподілити матеріали та завдання а також використати нові підходи. Значного поширення набули такі формати організації змішаного навчання як метод «перевернутого навчання» (flipped classroom) – ціллу уроків стає виконання практичних завдань та обговорення, тоді як ознайомлення із матеріалом відбувається вдома. Також часто використовують технологію BYOD (Bring Your Own Device), за якої учасники використовують на заняттях власні пристрої для інтерактивних завдань, часто, із використанням певного програмного забезпечення (Ребуха та ін., 2022). Варто зазначити, що у змішаному навчанні значну роль відіграють навчальні матеріали представлені в електронному вигляді.

Існує велика кількість програмних рішень для забезпечення електронного навчання, від таких, що покривають конкретну потребу, наприклад, засоби організації відеоконференцій для встановлення комунікації, до комплексних програмних систем, до яких можна віднести системи управління навчальним вмістом (LCMS Learning Content Management System), системи управління навчанням (LMS Learning Management System) (Топузов, 2017).

Системи управління навчанням надають інструменти для управління освітнім процесом, забезпечують можливості для розміщення матеріалів та ознайомлення із ними, надають середовище для виконання та оцінювання завдань, можуть також здійснювати збір та аналіз продуктивності та успішності учнів. Таким чином LMS дозволяють налагодити взаємодію між учнями, викладачами та адміністрацією освітньої установи (Barreto та ін., 2020).

Системи LCMS орієнтовані на створення та надання навчальних матеріалів у спеціальних форматах та модульному представленні, що забезпечує можливість повторного використання, якісного збереження та поширення контенту (Топузов, 2017).

Сьогодні комплексні системи дозволяють поєднувати можливості LMS для організації навчального процесу та LCMS для створення контенту та управління ним.

Необхідно також виконати огляд структури та принципів функціонування систем управління навчанням. Вони спрямовані на забезпечення потреб учнів, викладачів та адміністрації. Однак можуть залучись й інші користувачі на різних етапах розроблення та використання системи, наприклад, конструктори матеріалів, основна функція яких полягає у розробленні навчальних матеріалів. Відповідно, кожен користувач повинен мати достатній обсяг функціональних можливостей для ефективного використання системи.

В умовах асинхронних, частково або повністю віддалених взаємодій, важливо забезпечити учасників освітнього процесу інструментами для комунікації, до яких можна віднести форуми, що створюють простір для обговорень певних тем. Чати

дозволяють підтримувати зв'язок із викладачем для оперативного вирішення питань або проблем. Комунікаційні засоби сучасних систем управління навчанням не обмежуються лише згаданими інструментами, а можуть включати також інтерактивні дошки, вбудовані сервіси аудіо та відеоконференцій, засоби обміну файлами тощо (Arola & Aaltonen, 2023).

Системи управління навчанням можуть включати інструменти для адміністративного забезпечення процесу навчання, наприклад, розподілу функцій та управління записами користувачів, складання розкладів та планів, формування груп тощо.

Варто також згадати про інструменти для організації та ведення курсів. До них можна віднести засоби для створення структури курсу, формування та розміщення матеріалів, які можуть створюватись на основі шаблонів для економії часу та зусиль. Самі матеріали можуть бути представлені у різних форматах: зображення, графіки, відео, текст, таблиці, мультимедіа. Системи управління навчанням також дозволяють організовувати повний цикл роботи із завданнями, включаючи створення, призначення, визначення умов та термінів, перевірки та виставлення оцінки, надання відгуку зі сторони викладача а також виконання та надсиланням на перевірку студентом (Barreto та ін., 2020).

Необхідно наголосити на важливих додаткових функціях, які часто реалізовані у системах управління навчанням: календар для відслідковування термінів та розкладу, електронний журнал для ведення та відслідковування успішності учасників курсу.

Особливу роль у системах управління навчанням відіграють засоби тестування знань – програмне забезпечення для організації процесу тестування. Під поняттям тесту розуміють сукупність завдань стандартизованої форми, що створюється з метою оцінювання рівня знань особи. Тестування може застосовуватись на різних етапах навчального процесу: для оцінювання знань отриманих на певному занятті – поточне тестування, для закріплення певної теми – тематичне, підсумкове – для вимірювання якості опрацювання всього курсу. Тестування

дозволяє не лише визначити рівень знань особи, а й передбачити готовність до засвоєння подальших тем, систематизувати знання. На відміну від традиційних способів контролю, тестування забезпечує об'єктивність оцінювання, визначення повноти знань а також дозволяє економити час на проведення та подальше опрацювання результатів. Однак існує ряд недоліків, серед яких варто виділити складність у перевірці глибини знань та ходу думок особи, можливість вгадати відповідь а також значний обсяг часу необхідний для складання запитань та варіантів відповідей (Лузан та ін., 2022).

Процес тестування охоплює ряд етапів: планування, де визначають мету та тематику. Може входити також етап формування матриці тесту, де вказують цілі навчання, кількість запитань із кожної теми. Навчальні цілі можуть бути класифіковані різними способами, найбільш відомим серед яких є таксономія Блума із поділом на такі рівні: знання – просте запам'ятовування, розуміння, застосування, аналіз – знання структури, синтез – можливість поєднувати виокремлені елементи, оцінювання отриманого знання. Наступним кроком у процесі тестування є формування банку запитань після чого здійснюють саме тестування і опрацьовують його результати (Сергієнко & Кухар, 2011).

Існує велика кількість метрик та показників, за допомогою яких можна оцінити якість тесту та правильно інтерпретувати результати його проведення. Часто використовують поняття валідності тесту загалом та тестових завдань зокрема, що вказують на відповідність фактичних та очікуваних вимірювань. Застосовують також показники рівня легкості – відношення фактично набраної суми балів до максимально можливої, дискримінаційний індекс – наскільки якісно завдання може розподілити учасників за рівнем підготовки.

Програмні системи тестування знань можуть існувати як окремі рішення, так і у складі окремого модуля LMS. Вони можуть включати інструменти для створення банку запитань, безпосереднього проведення тестування а також опрацювання результатів. Системи тестування спрощують організацію тестування, розширюють можливості вчителів, дозволяють

подавати запитання та варіанти відповідей у випадковому порядку, додавати графічні та відеоматеріали, автоматизують процес оцінювання та інтерпретації результатів. Багато програмних засобів дозволяють отримувати звіти, зокрема у графічному вигляді про результати тестування, реалізовувати концепцію адаптивного тестування для можливості підлаштовуватись під рівень знань та потреб особи (Фетісов, 2011).

Також системи тестування надають можливість використовувати усі базові види тестових завдань, які можна розділити на дві групи: із закритою та відкритою формами відповіді. У першому випадку необхідно вибрати або упорядкувати один або декілька варіантів із запропонованих. У завданнях другої групи потрібно вказати власну відповідь, яка може бути представлена у вигляді числа, слова, речення тощо. Завдання закритої форми поділяють на завдання визначення істинності твердження, шляхом вибору із двох варіантів. До них також відносять завдання вибору однієї правильної відповіді серед запропонованих 4-5 варіантів, неправильні варіанти називають дистракторами, які повинні бути створені з урахуванням можливих помилок, яких може припуститись особа при вирішенні завдання. Варто згадати про завдання закритої форми із множинним вибором, де потрібно обрати декілька варіантів, кількість яких може бути вказаною або ні. Часто використовують завдання на встановлення послідовності а також відповідності, останні можуть існувати у декількох формах (Сергієнко & Кухар, 2011).

Багато сучасних систем управління навчанням дозволяють створювати та розповсюджувати матеріали із використанням стандартів представлення, найбільш відомими серед яких є SCORM (Shareable Content Object Reference Model) та xAPI. Завдяки використанню формату SCORM з'явилась можливість відслідковувати взаємодію із контентом та використовувати матеріали у різних системах управління навчанням. Впровадження стандарту xAPI, дозволило значно розширити можливості адаптивності та відслідковування активностей (Самойленко та ін., 2017).

Сьогодні існує та використовується велика кількість систем управління навчанням. Одними із найбільш поширених є Moodle та Canvas.

Система Moodle (Modular Object-Oriented Dynamic Learning Environment) надає безліч інструментів для організації навчального процесу та управління вмістом, зокрема виділяють адміністративний модуль (налаштування системи, керування правами доступу та ролями, зміна зовнішнього вигляду сторінок). Також надаються необхідні функціональні можливості для наповнення курсів та формування уроків: електронний журнал, чати, форуми, тести, сторінки, завдання тощо. Модуль тестування надає широкі можливості зі створення тестів, зокрема дозволяє визначати спосіб проходження та представлення тесту (Khatser, G., & Khatser, M., 2022; Moodle: the definitive guide to the LMS)

Система Canvas є доступною для безкоштовного використання для педагогів із обмеженим функціоналом. Вона пропонує дружній до користувача інтерфейс із зручною навігацією, дозволяє створювати матеріали із використанням текстового редактора, підтримує велику кількість розширень, наприклад, для аналітики, відслідковування активності, персоналізованого навчання. Також забезпечує учасників освітнього процесу засобами комунікації (чат, обмін повідомленнями, платформа для проведення дискусій), тестування із базовими типами запитань, електронного журналу із можливостями імпортування та експортування даних (Khatser, G., & Khatser, M., 2022) .

Порівняльний аналіз функціональних можливостей вищезазначених систем за рядом критеріїв представлено у таблиці 1.

Таблиця 1

Порівняння систем управління навчанням

Назва критерію	Назва системи управління навчанням	
	Moodle	Canvas
Тип	З відкритим вихідним кодом	Комерційна / з відкритим вихідним кодом
Вартість	Безкоштовна (наявні витрати за розгортання)	Безкоштовна для базового функціоналу, платна для розширеного
Можливості для здобувачів освіти	Панель інструментів та календар	Зручна панель інструментів, календар, швидка навігація
Модуль адміністрування	Досить широкі можливості для керування користувачами, правами доступу, системою.	Можливості керування користувачами та правами доступу
Підтримувані пристрої	Веб версія, мобільний застосунок	Веб версія, мобільний застосунок
Завдання та тестування	Підтримуються основні а також додаткові типи тестових питань, надається велика кількість інструментів для налаштування тесту	Підтримуються основні типи тестових запитань та базові налаштування процесу тестування
Розроблення курсів	Широкий спектр налаштувань, видів доступних ресурсів, однак може потребувати більше часу	Швидке та інтуїтивне створення, обмежені можливості налаштувань
Функції для педагогів	Створення курсів, розділів, завдань, тестів, матеріалів, швидке оцінювання, електронний журнал	Створення та наповнення курсів, додавання матеріалів, завдань, швидке оцінювання, зручний журнал із великою кількістю функцій
Інтеграція	Можливість додавання пакетів SCORM, AICC та IMS а також підключення програмних модулів	Можливість додавання пакетів SCORM а також деяких інших постачальників, якісні інтеграції від відомих партнерів
Комунікація	Наявні інструменти для особистої та групової комунікації з гнучкими налаштуваннями	Широкий спектр засобів для комунікації, відзначається швидкість та якість взаємодій

У результаті порівняння систем управління навчанням, можна зазначити, що Moodle пропонує більшу кількість налаштувань, але можуть знадобитися додаткові зусилля для якісного конфігурування. Canvas LMS пропонує меншу гнучкість у налаштуваннях, однак забезпечує користувачів інтуїтивно зрозумілим інтерфейсом та легко інтегрується із додатковими сервісами для розширення можливостей.

Було також визначено, що жодна із вищезгаданих систем управління навчанням не має вбудованих рішень для рекомендацій подальших курсів, хоча обидва рішення мають деякі можливості для інтеграції з використанням додаткових інструментів. Таким чином виникає розрив у навчанні та труднощі у пошуку напрямків розвитку. Відтак, доцільно забезпечити користувачів інструментами для підбору наступних курсів на основі визначених інтересів.

Програмні системи навчання та тестування знань можуть бути представлені у різних формах, однак було обрано підхід до створення у вигляді вебзастосунка через велику кількість переваг такого підходу. Під поняттям вебзастосунка розумітимемо застосунок із клієнт-серверною архітектурою, де клієнт відповідає за надання користувачеві інтерфейсу для взаємодії, збору та відображення інформації, тоді як сервер відповідає за обробку запитів та роботу з даними. Досить поширеною та ефективною сьогодні є архітектура, що складається із трьох рівнів: презентаційного – відповідає за взаємодію з користувачем, рівня застосунків – реалізація бізнес-логіки та рівня даних – доступ та збереження даних з використанням систем управління базами даних (Understanding the architecture..., 2024).

Вирізняють декілька видів вебзастосунків: MPA (Multi-page application) – багатосторінкові вебзастосунки, які для кожного запиту завантажують нову сторінку. SPA (Single Page Application) – односторінкові вебзастосунки при першому зверненні завантажують сторінку і при подальших запитах оновлюють лише необхідні блоки, внаслідок чого значно підвищується якість користувацького досвіду у зв'язку із більш швидким відображенням результату. Також варто згадати про PWA

(Progressive Web Application) – прогресивний вебзастосунок, що розширює можливості звичайних рішень та дозволяє після встановлення отримувати доступ до частини функціоналу без підключення до мережі Інтернет, однак за такого підходу можуть виникати проблеми із сумісністю з деякими браузерами (PWA vs. MPA vs. SPA..., 2025).

Для забезпечення комунікації між частинами вебзастосунка розробляють та використовують API (Application Programming Interface) – набір інструкцій, що дозволяє компонентам програми взаємодіяти. Доволі поширеною практикою для організації взаємодії між клієнтською та серверною частинами є використання REST API (Representational State Transfer API), який пропонує декілька принципів для створення якісних та зрозумілих API, серед яких використання клієнт-серверної архітектури, відсутність стану, надання повноти інформації у запиті для обробки.

Для реалізації клієнтської частини вебзастосунка було вирішено обрати такий набір технологій: мову розмітки HTML (Hypertext Markup Language), що дозволяє визначити структуру сторінки за допомогою тегів, які описують спосіб відображення вмісту. Для стильового оформлення сторінок було обрано мову CSS (Cascading Style Sheets) із препроцесором SCSS (Sassy Cascading Style Sheets), що значно полегшує складні оформлення блоків. Для надання динамічності та обробки клієнтської взаємодії було використано інтерпретовану, динамічно-типізовану мову програмування JavaScript. Існує багато фреймворків, які дозволяють значно спростити процес розроблення, серед яких одним із найбільш поширених є ReactJS. Це компонентно-орієнтована бібліотека, що дозволяє будувати інтерфейс із окремих повторно використовуваних блоків, а також забезпечує рівень абстракції над DOM (Document Object Model) (Прокопчук, 2024).

Також охарактеризуємо вибрані технології для реалізації серверної частини вебзастосунку. У ході створення рівня бізнес-логіки було використано мову програмування Java у поєднанні з фреймворком Spring. Spring Framework дозволяє значно спростити створення вебзастосунків, зокрема надає механізми

інверсії управління (Inversion of Control) та впровадження залежностей, які забезпечують незалежність компонентів програми та легке їх поєднання за потреби. Велику кількість необхідного функціоналу пропонують інші проекти Spring, наприклад, Spring Boot – фреймворк, що слідує принципу «конвенція замість конфігурації» та надає велику кількість стандартних інструментів для швидкого розроблення програмного забезпечення, таких як автоматична конфігурація, вбудований сервер, автоматичне управління пакетами. Таким чином додаткові налаштування необхідні лише за нестандартних умов (Spring vs Spring Boot, 2024). Spring Data забезпечує зручну інтеграцію з базами даних та реалізує принцип об'єктно-реляційного відображення.

Враховуючи специфіку предметної області та рівень структурованості даних програмної системи, було вирішено обрати реляційну модель баз даних. Як систему управління базами даних було використано PostgreSQL, яка є поширеною сьогодні у проєктах різних масштабів, завдяки відкритості, підтримці широкого спектру типів даних, великій кількості підтримуваних платформ та високому рівню відповідності принципам ACID (Atomicity, Consistency, Isolation, Durability).

Безпека даних у програмних системах навчання та тестування знань є ще одним аспектом, який потрібно враховувати у ході розроблення. Важливо забезпечити надійні процедури автентифікації та авторизації користувача, паролі повинні бути довгими а зберігати їх необхідно винятково у перетвореному за допомогою хеш-функції вигляді. Після успішної автентифікації взаємодію між клієнтом та сервером організовують за допомогою JWT-токенів (JSON Web Token), які містять необхідну для ідентифікації користувача інформацію. Зазвичай використовують два види токенів access – із коротким терміном дії, що використовується для доступу до захищених ресурсів, а також refresh – із достатньо довгим терміном дії, для отримання нового access-токена. Таким, чином значно полегшується взаємодія користувача із застосунком, а використання алгоритму хешування гарантує незмінність самого токена (Garg, 2024). Засобом який може значно спростити згадані процеси є

фреймворк Spring Security, робота якого базується на використанні ланцюга фільтрів (Filter Chain), що містить обробники для різних сценаріїв (Любчак та ін., 2023).

Реалізація модуля рекомендацій курсів може бути виконана шляхом аналізування тематики матеріалів та визначення ключових слів та словосполучень, які характеризуватимуть текст. Статистичні методи визначення ключових слів за частотою появи не надають належного рівня точності. Тому для підвищення якості визначення та виявлення семантично близьких ключових слів до тексту було використано технології машинного навчання. Зокрема моделі типу BERT (Bidirectional Encoder Representations from Transformers) базуються на двонаправлених трансформерах та дозволяють представляти тексти у формі векторів на основі використання мовних моделей із врахуванням контексту (Khan та ін., 2022; Amur та ін., 2023). Реалізацією такого підходу є бібліотека KeyBERT для мови програмування Python, що дозволяє на основі попередньо навчених моделей BERT, визначати слова-кандидати та аналізувати подібність цих слів у відношенні до самого тексту методом косинусової подібності (Giarelis та ін., 2021). Такі інструменти як Stanza, що виконує попередню синтаксичну обробку тексту, дозволяють підвищити якість визначення ключових слів. Отримані ключові слова у результаті роботи моделі дозволяють характеризувати тематику курсу а також інтереси студентів, на основі успішності та курсів, що обираються у процесі навчання. Далі одержана інформація застосовується для пошуку семантично близьких курсів, зважаючи на уподобання.

На основі розглянутих теоретичних основ та підходів до створення програмних систем навчання та тестування знань, виконаного порівняльного аналізу наявних рішень було сформовано вимоги до розроблюваної системи та представлено у вигляді UML-діаграми на рисунку 1.

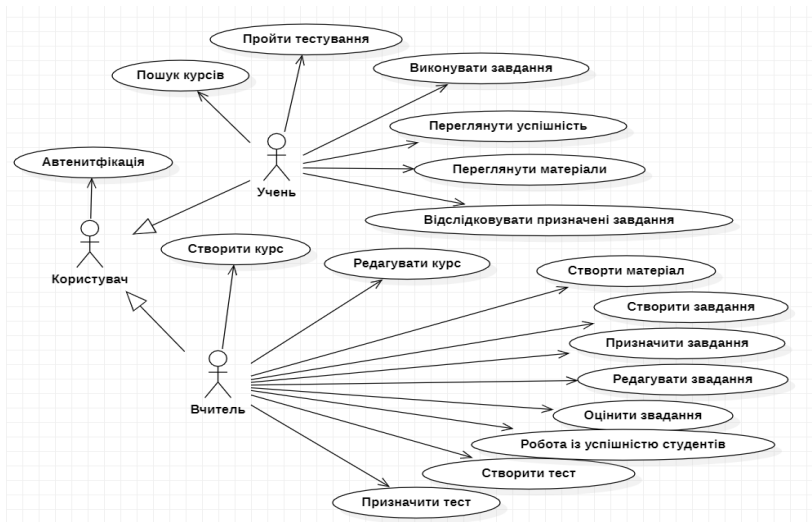


Рис. 1. Діаграма варіантів використання

Були визначені основні сутності та зв'язки, що виникають між ними: сутність Користувача дозволяє описати особу у системі, визначити її ролі та права доступу. Сутність учня дозволяє описати користувача, що використовує її з ціллю отримати нові знання. Вчитель використовує систему для створення курсів, завдань, матеріалів та тестів, ведення викладацької діяльності, проведення оцінювання. Введено сутність курсу, що може містити завдання, уроки та тести, створюється вчителем, який долучає до проходження учнів. Уроки мають модульну структуру та дозволяють забезпечити учнів навчальними матеріалами. Завдання створюються вчителями та призначаються до виконання учням для подальшого оцінювання. Сутність тесту містить запитання та відповіді, створюється у межах курсу та призначається для проходження учням.

Важливим етапом у ході розроблення системи є тестування програмного забезпечення, що дозволяє запобігти неочікуваній поведінці, виявити критичні вразливості та помилки у функціонуванні системи. Було створено Unit-тести мовою Java із використанням фреймворка JUnit. Також використано підхід

мануального тестування (manual tests) для перевірки працездатності системи на більш високому рівні.

На основі сформованих вимог було реалізовано програмну систему. Розглянемо функціональні можливості більш детально. На початку взаємодії користувач потрапляє на головну сторінку, де відображаються відкриті для реєстрації курси. Для повноцінної взаємодії необхідно увійти в обліковий запис або створити новий, після чого буде отримано повноцінний доступ до системи. Головна сторінка в такому випадку буде представлена трьома основними вкладками: доступних для реєстрації курсів із можливістю пошуку, тих, на які зареєстрований учень, а також буде надано можливість переглядати рекомендовані подібні курси в окремому розділі. Функціонал вчителя орієнтований на створення, редагування та роботу із курсами, тому він може додавати нові курси та переглядати курси, які веде.

Після того як учень обирає певний курс відображається набір вкладок, зокрема загальні, де розміщені розділи для уроків із переліком сторінок матеріалів, завдань – список завдань із короткою інформацією (оцінка, терміни виконання), а також тестів із можливістю почати проходження заданих тестів. Є вкладка загальної інформації про курс, де вказано опис, терміни та викладачів курсу, також доступна вкладка оцінок, де можна ознайомитись із успішністю.

Викладач має схожий інтерфейс для взаємодії з курсами, є додаткові можливості додавання нових матеріалів та редагування створених. Реалізовано функції додаткового редагування інформації по курсу на вкладці загальних матеріалів, перегляду успішності студентів у розділі оцінок. Також є окрема вкладка Студенти, що дозволяє додавати та вилучати студентів.

У разі обрання конкретного уроку учень може переглянути матеріали представлені у вигляді зображень, тексту, відео, або завантажити прикріплений файл. Вчитель має додаткові можливості створення та редагування із використанням блоків для контенту певного виду, зокрема текстового редактора для форматування тексту.

При переході до обраного тесту інтерфейс відрізнятиметься для учня та вчителя. Учнію буде запропоновано розпочати проходження тестування, після чого будуть відображені запитання та таймер. Питання можуть бути представлені по одному на сторінці або увесь перелік одразу. Після завершення тестування з'явиться можливість переглянути кількість набраних балів та додатково правильність виконання кожного із завдань у разі обрання вчителем відповідних налаштувань.

Вчитель має можливість створювати нові тести, визначати порядок у якому будуть відображені запитання, формувати завдання різних видів: з однією чи декількома правильними відповідями, встановлення послідовності або відповідності, на вказання правильної відповіді власноруч. На рисунку 2 зображено сторінку створення тесту, а саме блок створення запитання, тут можна вказати умову використовуючи текстовий редактор, кількість балів за запитання, варіанти відповіді а також позначити правильний.

The screenshot shows the 'StudyIt!' interface for creating a question. At the top, there's a blue header with the 'StudyIt!' logo and navigation icons. The main content area is titled 'Заяпитання' (Create Question). It features a form with the following elements:

- A title field with the placeholder 'Бали за запитання' (Points for question) and a value of '1'.
- A text editor for the question text, with a toolbar showing options like font size, bold, italic, underline, and list creation. The text inside the editor is 'Яка із наведених мов програмування НЕ є об'єктно-орієнтованою?' (Which of the listed programming languages is NOT object-oriented?).
- A section for 'Варіанти відповідей' (Answer options) with three input fields containing 'Java', 'Haskell', and 'C++'. Each field has a radio button to its right, labeled 'Правильний' (Correct). The 'Haskell' option is selected.
- A 'Додати варіант' (Add option) button at the bottom of the options list.
- A blue button labeled 'ОБРАТИ ТИП' (Select type) at the bottom of the form.
- A link labeled 'СТВОРИТИ ТЕСТ' (Create test) below the main form.

Рис. 2. Сторінка створення тесту

Щодо завдань, то учень може ознайомитись із інструкцією виконання, що також може бути представлена різними видами подання інформації, термінами, прикріпити результати

виконання та надіслати роботу на перевірку, ознайомитись із отриманою оцінкою. Викладач має можливість створювати завдання, встановлювати вимоги до виконання, визначати для кого із студентів воно призначене, здійснювати перевірку та оцінювання робіт учнів.

Також для вчителів та учнів є можливість переглядати календар, де відображені терміни виконання завдань та тестів, що значно спрощує планування часу.

Дискусія і висновки

У результаті виконаного дослідження було досягнуто поставленої мети та виконано усі завдання: досліджено теоретичні основи щодо програмних систем навчання та тестування знань, виконано аналіз програмно-технологічних рішень для побудови програмних систем навчання та тестування знань, здійснено проектування, реалізацію та тестування програмної системи навчання та тестування знань, розроблено та інтегровано модуль рекомендацій курсів на основі уподобань студентів. Розроблена система дозволяє оптимізувати процес навчання та тестування знань та підвищити рівень гнучкості для учасників освітнього процесу та може бути використана у закладах освіти та організаціях, які надають освітні послуги.

Подальші дослідження можуть бути спрямовані на додавання елементів гейміфікації, інтеграції з іншими сервісами а також впровадження технологій штучного інтелекту для глибшої автоматизації, персоналізації та адаптивності навчання.

Список використаних джерел

Гулай, О., Кабак, В., & Герасимчук, Г. (2023). *Засоби та технології цифрового навчання: теоретичний та практичний аспекти: монографія*. ЛНТУ.

Лузан, П., Лапа, О., Пашенко, Т., Мося, І., Ваніна, Н., Остапенко, А., & Ямковий, О. (2022). *Методичні рекомендації щодо розроблення валідних тестів у закладах фахової передвищої освіти*. Інститут професійної освіти НАПН України. <https://doi.org/10.32835/978-61795280-5-7/2022>

Любчак, В. О., Савостян, В. В., & Козолуп, П. Д. (2023). *Безпека Java-додатків: навчальний посібник*. Сумський державний університет.

Прокопчук, Ю. (2024). Використання бібліотеки React для розроблення вебзастосунків: переваги та виклики. У Прикладні інформаційні системи та технології в цифровому суспільстві (с. 262–267).

Ребуха, Л. З., Білоус, І. І., Брик, Р. С., Дем'янюк, А. В., Савіцька, В. В., Кричківська, О. В., & Кавецький, В. Є. (2022). *Інноваційні технології навчання в умовах модернізації сучасної освіти: монографія*. ЗУНУ. <http://dspace.wupn.edu.ua/handle/316497/48105>

Самойленко, О. М., Бацуровська, І. В., Ручинська, Н. С., & Самойленко, О. О. (2017). *Технологічні характеристики систем управління навчанням*. World Science : International Scientific and Practical Conference (с. 60–64).

Сергієнко, В., & Кухар, Л. (2011). *Методичні рекомендації зі складання тестових завдань*. НПУ ім. М.П. Драгоманова.

Топузов, М. О. (2017). *Проектування інформаційно-освітнього середовища навчальних закладів у сучасному суспільстві*. Український педагогічний журнал, (1), (с. 26–36).

Фетисов, В. (2011). *Комп'ютерні технології в тестуванні*. ПП Лисенко М.М. Amur, Z. H., Hooi, Y. K., Soomro, G. M., Bhanbhro, H., Karyem, S., & Sohu, N. (2023). *Unlocking the Potential of Keyword Extraction: The Need for Access to High-Quality Datasets*. Applied Sciences, 13(12), 7228. <https://doi.org/10.3390/app13127228>

Arola, T., & Aaltonen, S. (2023). *Manual for e-learning design*. Uniwersytet Ekonomiczny we Wrocławiu.

Barreto, D., Rottmann, A., & Rabidoux, S. (2020). *Learning management systems choosing the right path for your organization*. Ed Tech Books. https://edtechbooks.org/learning_management_systems

Cleveland-Innes, M., & Wilton, D. (2018). *Guide to blended learning*. Commonwealth of Learning.

Garg, R. (2024, 6 липня). *Significance of a JWT Refresh Token*. Baeldung. <https://www.baeldung.com/cs/json-web-token-refresh-token>

Giarelis, N., Kanakaris, N., & Karacapilidis, N. (2021). *A Comparative Assessment of State-Of-The-Art Methods for Multilingual Unsupervised Keyphrase Extraction*. У IFIP Advances in Information and Communication Technology (с. 635–645). Springer International Publishing. https://doi.org/10.1007/978-3-030-79150-6_50

Khan, M. Q., Shahid, A., Uddin, M. I., Roman, M., Alharbi, A., Alosaimi, W., Almalki, J., & Alshahrani, S. M. (2022). *Impact analysis of keyword extraction using contextual word embedding*. PeerJ Computer Science, 8, Стаття e967. <https://doi.org/10.7717/peerj-cs.967>

Khatser, G., & Khatser, M. (2022). *Online Learning Through LMSs: Comparative Assessment of Canvas and Moodle*. International Journal of Emerging Technologies in Learning (iJET), 17(12), pp. 184–200. <https://doi.org/10.3991/ijet.v17i12.30999>

Moodle: the definitive guide to the LMS. (2022, 11 листопада). SupportHost. <https://supporthost.com/moodle/>

PWA vs. MPA vs. SPA - What's the Best Choice for Your App? (2024, 30 жовтня). Neoteric — Your Tech Partner for Software Development and AI. <https://neoteric.eu/blog/pwa-vs-mpa-vs-spa-whats-the-best-choice-for-your-app>

Spring vs Spring Boot: An In-depth Comparison - Tech Talents and Innovation. (2024, 18 квітня). Ello Talent: website. <https://ello.io/spring-vs-spring-boot/>

Understanding the architecture of a 3-tier application. (2024, 13 травня). vFunction. <https://vfunction.com/blog/3-tier-application>

References

Amur, Z. H., Hooi, Y. K., Soomro, G. M., Bhanbhro, H., Karyem, S., & Sohu, N. (2023). *Unlocking the Potential of Keyword Extraction: The Need for Access to High-Quality Datasets*. Applied Sciences, 13(12), 7228. <https://doi.org/10.3390/app13127228>

Arola, T., & Aaltonen, S. (2023). *Manual for e-learning design*. Uniwersytet Ekonomiczny we Wrocławiu.

Barreto, D., Rottmann, A., & Rabidoux, S. (2020). *Learning management systems choosing the right path for your organization*. Ed Tech Books. https://edtechbooks.org/learning_management_systems

Cleveland-Innes, M., & Wilton, D. (2018). *Guide to blended learning*. Commonwealth of Learning.

Fetisov, V. (2011). *Computer technologies in testing*. PE Lysenko M.M [in Ukrainian].

Garg, R. (2024, July 6). *Significance of a JWT Refresh Token*. Baeldung. <https://www.baeldung.com/cs/json-web-token-refresh-token>

Giarelis, N., Kanakaris, N., & Karacapilidis, N. (2021). *A Comparative Assessment of State-Of-The-Art Methods for Multilingual Unsupervised Keyphrase Extraction*. Y IFIP Advances in Information and Communication Technology (с. 635–645). Springer International Publishing. https://doi.org/10.1007/978-3-030-79150-6_50

Gulay, O., Kabak, V., & Gerasymchuk, H. (2023). *Digital learning tools and technologies: theoretical and practical aspects: monograph*. LNTU [in Ukrainian].

Khan, M. Q., Shahid, A., Uddin, M. I., Roman, M., Alharbi, A., Alosaimi, W., Almalki, J., & Alshahrani, S. M. (2022). *Impact analysis of keyword extraction using contextual word embedding*. PeerJ Computer Science, 8, Стаття e967. <https://doi.org/10.7717/peerj-cs.967>

Khatser, G., & Khatser, M. (2022). *Online Learning Through LMSs: Comparative Assessment of Canvas and Moodle*. International Journal of Emerging Technologies in Learning (iJET), 17(12), pp. 184–200. <https://doi.org/10.3991/ijet.v17i12.30999>

Luzan, P., Lapa, O., Pashchenko, T., Mosya, I., Vanina, N., Ostapenko, A., & Yamkovy, O. (2022). *Methodological recommendations for developing valid tests in higher education institutions*. Institute of Professional Education of the National Academy of Pedagogical Sciences of Ukraine. <https://doi.org/10.32835/978-61795280-5-7/2022> [in Ukrainian]

Lyubchak, V. O., Savostyan, V. V., & Kozolup, P. D. (2023). *Java application security: a training manual*. Sumy State University [in Ukrainian].

Moodle: the definitive guide to the LMS. (2022, November 11). SupportHost. <https://supporthost.com/moodle/>

Prokopchuk, Y. (2024). *Using the React library for web application development: advantages and challenges*. In Applied Information Systems and Technologies in the Digital Society (pp. 262–267).

PWA vs. MPA vs. SPA - What's the Best Choice for Your App? (2024, October 30). Neoteric — Your Tech Partner for Software Development and AI. <https://neoteric.eu/blog/pwa-vs-mpa-vs-spa-whats-the-best-choice-for-your-app>

Rebukha, L. Z., Bilous, I. I., Bryk, R. S., Demianuk, A. V., Savitska, V. V., Krychivska, O. V., & Kavetskyi, V. Ye. (2022). *Innovative teaching technologies in the context of modernizing contemporary education: monograph*. WUNU. <http://dspace.wunu.edu.ua/handle/316497/48105> [in Ukrainian].

Samoilenko, O. M., Batsurovska, I. V., Ruchynska, N. S., & Samoilenko, O. O. (2017). *Technological characteristics of learning management systems*. World Science: International Scientific and Practical Conference (pp. 60–64) [in Ukrainian].

Sergienko, V., & Kukhar, L. (2011). Methodological recommendations for compiling test tasks. M.P. Dragomanov National Pedagogical University [in Ukrainian].

Spring vs Spring Boot: An In-depth Comparison - Tech Talents and Innovation. (2024, April 18). Ellow Talent: website. <https://ellow.io/spring-vs-spring-boot/>

Topuzov, M. O. (2017). *Designing the information and educational environment of educational institutions in modern society*. Ukrainian Pedagogical Journal, (1), (pp. 26–36) [in Ukrainian].

Understanding the architecture of a 3-tier application. (2024, May 13). vFunction. <https://vfunction.com/blog/3-tier-application>

Отримано редакцією журналу / Received: 26.09.25

Прорецензовано / Revised: 27.09.25

Схвалено до друку / Accepted: 01.10.25

Yurii PROKOPCHUK, Student
ORCID ID: 0009-0008-7363-2080
e-mail: yuriiprokopchuk@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Olena FENDYO, PhD (Tech.), Assoc. Prof.
ORCID ID: 0000-0001-9197-6399
e-mail: olena1.fendyo@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

SOFTWARE SYSTEM FOR LEARNING AND KNOWLEDGE TESTING

The article considers approaches to creating a software system for learning and knowledge testing aimed at increasing the efficiency of the learning process in the context of the education digital transformation. An analysis of modern theoretical and methodological foundations of e-learning and e-testing is performed, the features of learning management systems (LMS) and technological solutions used in their development, implementation and testing are considered. A comparative analysis of existing learning management systems was carried out using Moodle and Canvas as examples. An architecture is proposed and a web application that automates the processes of learning, control, and knowledge assessment is implemented using modern frontend (React, JavaScript) and backend (Java, Spring Boot, PostgreSQL) development technologies. A module for course recommendations based on user preferences is created using the Python programming language and the KeyBERT library. The results confirm the possibility of effective use of the developed web system in educational institutions and organizations that provide educational services.

Key words: e-learning, e-testing, learning management system, educational software system, LMS.

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.